MCL-NetS24 Version 2

User's Guide

MCL-Net S24 Version 2

User's Guide

70-33348-02

Revision A

June 1999

symbol®

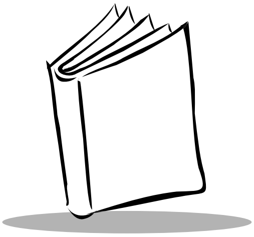
Symbol reserves the right to make changes to any software or product to improve reliability, function, or design.

Symbol does not assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

No license is granted, either expressly or by implication, estoppel, or otherwise under any Symbol Technologies, Inc., intellectual property rights. An implied license only exists for equipment, circuits, and subsystems contained in Symbol products.

Symbol, Spectrum One, and Spectrum24 are registered trademarks of Symbol Technologies, Inc. MCL is a registered trademark and MCL-Designer, MCL-Link, MCL-Loader, and MCL-Net S24 are trademarks of Zetes Technologies S.A. Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Symbol Technologies, Inc.
One Symbol Plaza
Holtsville, New York 11742-1300
http://www.symbol.com



Contents

About This Guide

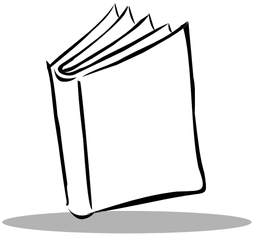
Chapter 1. Using MCL-Net S24

Chapter 2. MCL-Net S24 DLL

Chapter 3. MCL-Net S24 Traffic Monitor

Index

Feedback



About This Guide

The *MCL-Net S24 User's Guide* provides general information about operating the MCL-Net S24 application, configuring the software, and using the MCL-Net S24 DLL.

The MCL-Net S24 is a 32-bit RF (Spectrum24) client/server communication application that:

- receives and saves data coming from the terminal
- manages all the file server functionality
- transfers files and MCL-Code programs to the terminals
- manages auxiliary functions like reset, date and time settings of the terminal.

System Requirements

MCL-Net S24 runs on a 486 computer running Windows 95 or Windows NT with at least 16 MB RAM, an Ethernet adapter card, TCP/IP drivers, and 10 MB of free hard drive space. A 133 MHz processor and a 16-bit color VGA monitor are recommended.

What's New in Version 2

MCL-Net S24 includes a powerful ODBC/SQL which allows direct interface with all ODBC compliant databases. ODBC technology provides an interface for accessing SQL databases. ODBC is based on SQL as a standard for accessing data. New Features in Version 2 include:

- Data dispatcher and load balancing configurer.
- New configuration and setup interface.
- Traffic monitoring and status.
- Windows NT compatible.
- Improved file transfer protocol with RF terminals.

Notational Conventions

The following conventions are used in this document:

- "Operator" and "User" refer to anyone using the MCL-NetS24 software.
- "PC" refers to the IBM personal computer or compatible system that you are using to develop applications.
- "Terminal" refers to various types of Symbol terminals.
- "You" refers to the administrator or person who is using this guide as a reference aid to install, configure, and/or operate the software.
- Keystrokes in bold type indicate non-alphanumeric keystrokes. For example: Select the <**F1**> key on the terminal to access on-line help.
- **Bold** type is used to identify menu items and input or text fields on a terminal screen.
- *Italics* are used:
  - for the names of parameters in function prototypes and variable names in usage and syntax descriptions
  - to highlight specific items in the general text
  - to identify chapters and sections in this and related documents.
- Square brackets [ ] in a command line enclose optional command line parameters.
- The piping symbol | has the effect of "or" when it is used to separate inline parameters on a command line; i.e., it separates alternative values for parameters.
- Bullets (•) indicate:
  - action items
  - lists of alternatives
  - lists of required steps that are not necessarily sequential.
- Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.

Related Publications

The following is a list of documents that you may find useful if you want to know more about the MCL-Collection.

- *MCL-Loader User's Guide*
  p/n 70-33347-xx

- *MCL-Link User's Guide*
  p/n 70-33346-xx
- *MCL-Designer User's Guide*
  p/n 70-33345-xx

Service Information

If you have a problem with your equipment, contact the *Symbol Support Centers*. Before calling, have the model number, serial number, and several of your bar code symbols at hand.

Call the Support Center from a phone near the scanning equipment so that the service person can try to talk you through your problem. If the equipment is found to be working properly and the problem is symbol readability, the Support Center will request samples of your bar codes for analysis at our plant.

If your problem cannot be solved over the phone, you may need to return your equipment for servicing. If that is necessary, you will be given specific directions.

Note: *Symbol Technologies is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty. If the original shipping container was not kept, contact Symbol to have another sent to you.*

Symbol Support Centers

For service information, warranty information or technical assistance contact or call the Symbol Support Center in:

United States

Symbol Technologies, Inc.
One Symbol Plaza
Holtsville, New York 11742-1300
1-800-653-5350

Canada

Symbol Technologies Canada, Inc.
2540 Matheson Boulevard East
Mississauga, Ontario, Canada L4W 4Z2
905-629-7226

United Kingdom

Symbol Technologies
Symbol Place
Winnersh Triangle, Berkshire RG41 5TP
United Kingdom
0800 328 2424 (Inside UK)
+44 118 945 7529 (Outside UK)

Asia/Pacific

Symbol Technologies Asia, Inc.
230 Victoria Street #04-05
Bugis Junction Office Tower
Singapore 188024
337-6588 (Inside Singapore)
+65-337-6588 (Outside Singapore)

Australia

Symbol Technologies Pty. Ltd.
432 St. Kilda Road
Melbourne, Victoria 3004
1-800-672-906 (Inside Australia)
+61-3-9866-6044 (Outside Australia)

Austria

Symbol Technologies Austria GmbH
Prinz-Eugen Strasse 70
Suite 3
2.Haus, 5.Stock
1040 Vienna, Austria
1-505-5794 (Inside Austria)
+43-1-505-5794 (Outside Austria)

Denmark

Symbol Technologies AS
Gydevang 2,
DK-3450 Allerod, Denmark
7020-1718 (Inside Denmark)
+45-7020-1718 (Outside Denmark)

Europe/Mid-East Distributor Operations

Contact your local distributor or call
+44 118 945 7360

Finland

Oy Symbol Technologies
Kaupintie 8 A 6
FIN-00440 Helsinki, Finland
9 5407 580 (Inside Finland)
+358 9 5407 580 (Outside Finland)

France

Symbol Technologies France
Centre d'Affaire d'Antony
3 Rue de la Renaissance
92184 Antony Cedex, France
01-40-96-52-21 (Inside France)
+33-1-40-96-52-50 (Outside France)

Germany

Symbol Technologies GmbH
Waldstrasse 68
D-63128 Dietzenbach, Germany
6074-49020 (Inside Germany)
+49-6074-49020 (Outside Germany)

Latin America Sales Support

7900 Glades Road
Suite 340
Boca Raton, Florida 33434 USA
1-800-347-0178 (Inside United States)
+1-561-483-1275 (Outside United States)

Netherlands

Symbol Technologies
Kerkplein 2, 7051 CX
Postbus 24 7050 AA
Varsseveld, Netherlands
315-271700 (Inside Netherlands)
+31-315-271700 (Outside Netherlands)

South Africa

Symbol Technologies Africa Inc.
Block B2
Rutherford Estate
1 Scott Street
Waverly 2090 Johannesburg
Republic of South Africa
11-4405668 (Inside South Africa)
+27-11-4405668 (Outside South Africa)

Italy

Symbol Technologies Italia S.R.L.
Via Cristoforo Columbo, 49
20090 Trezzano S/N Navigilo
Milano, Italy
2-484441 (Inside Italy)
+39-02-484441 (Outside Italy)

Mexico

Symbol Technologies Mexico Ltd.
Torre Picasso
Boulevard Manuel Avila Camacho No 88
Lomas de Chapultepec CP 11000
Mexico City, DF, Mexico
5-520-1835 (Inside Mexico)
+52-5-520-1835 (Outside Mexico)

Norway

Symbol Technologies
Trollasveien 36
Postboks 72
1414 Trollasen, Norway
66810600 (Inside Norway)
+47-66810600 (Outside Norway)

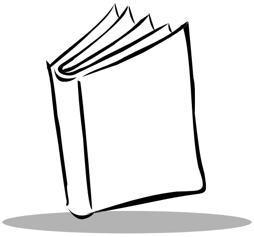
Spain

Symbol Technologies S.A.
Edificioi la Piovera Azul
C. Peonias, No. 2 - Sexta Planta
28042 Madrid, Spain
9-1-320-39-09 (Inside Spain)
+34-9-1-320-39-09 (Outside Spain)

Sweden

Symbol Technologies AB
Albygatan 109D
Solna
Sweden
84452900 (Inside Sweden)
+46 84452900 (Outside Sweden)

If you purchased your Symbol product from a Symbol Business Partner, contact that Business Partner for service.



Chapter 1
Using MCL-Net S24

Introduction

MCL-Net S24 is part of the MCL Collection of software. It can be installed separately or as part of the whole MCL Collection series. MCL-NetS24 is a Windows NT/95/98 communications server designed to work with Symbol’s character based Spectrum24 terminals running applications created using MCL-Designer. It supports up to 254 RF devices for one server.

MCL-NetS24 controls a variety of vital activities across the wireless network. It provides a flexible communications environment to interface with transaction-driven data collection applications.

Using MCL-NetS24, the Spectrum24 devices can upload or download data packets, files or programs anywhere within a wireless network range. Users can append, copy, rename or delete files on the server from a remote location. MCL-NetS24 provides time and date synchronization of wireless devices as well as a variety of interfaces to host applications through a Dynamic Link Library (DLL), files or pipes. MCL-NetS24 also includes a ODBC/ SQL engine that allows to the user to access and perform SQL request on ODBC compliant databases. MCL-NetS24 traces network activity, checks wireless terminals and Access Point status on the Ethernet and Spectrum24 networks, and manages both data paths and network devices.

Operational data may be easily saved on networked systems or drives by making declarations in MCL-NetS24 to set up the paths and destinations of data records and files. MCL-NetS24 constructs a Station ID configuration list that includes Access Point, Hosts and wireless devices. Using the master list of all wireless devices connected to the MCL-NetS24 communications server, enabling or disabling any device on the network becomes quick and easy. Spectrum24 activity may be traced for devices identified in the Station ID configuration list and this trace data can be viewed on a screen or written to a file for review. Interfacing MCL-Net24 communication programs are possible through Files, executable programs or DLLs.

Installation

To install MCL-Net S24:

1. Insert the MCL-Collection CD-ROM disk into the CD-ROM drive.
2. Click the Start button on the task bar.
3. Select Run.
4. Type **D:\setup.exe**. and click OK (replace **D:** with the CD-ROM drive letter).
5. Click Next when the introductory screen appears.
6. Click Yes to accept the license agreement.
7. When prompted, type your name and company in the appropriate fields. Click Next.
8. Click Next to accept the default directory or click Browse to select a different directory. Click OK, then Next.
9. Click Next to install all the MCL programs, or select MCL Net S24, Using DLL - C++ Sampler, and Using DLL - Visual Basic 5 Sampler and de-select all the other options.
10. Click Next to accept the MCL2 program folder.
11. Setup installs the selected programs onto the hard drive. A screen that monitors the progress appears. When installation is complete, click Finish.

MCL-Net S24

The main functionality of MCL-Net S24 program:

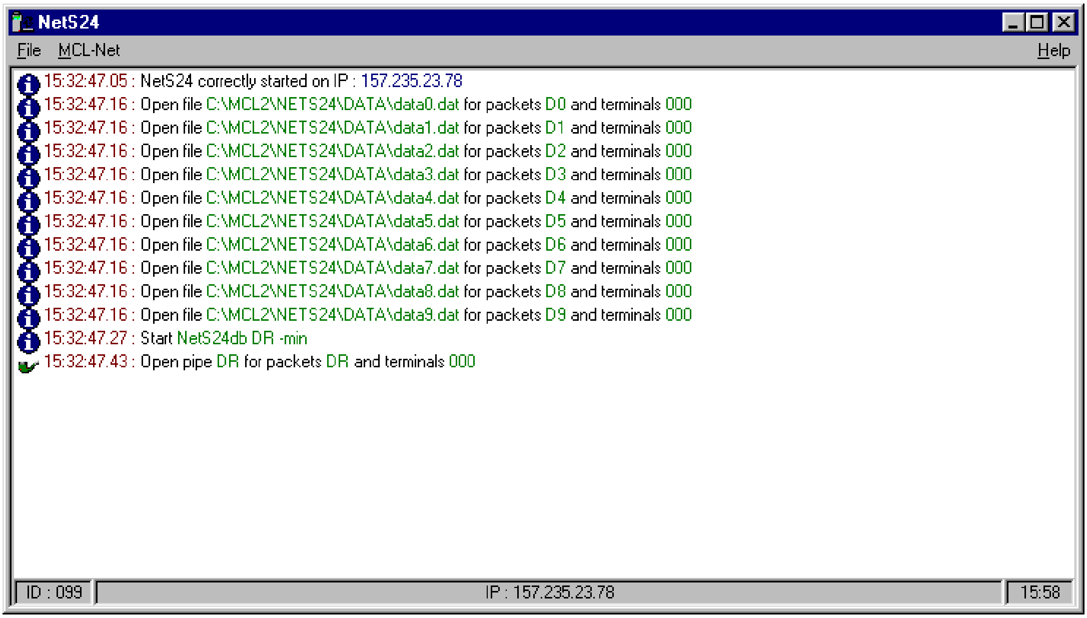
- receive and save data coming from terminals
- manage all “file server” functionality
- transfer files and MCL-Code programs to terminals
- manage auxiliary functions like reset, date and time setting of terminals.

Using MCL-Net S24

Starting MCL-Net S24

In oder to use the MCL-NetS24 program, a hardware key (dongle) must be installed on the host computer's parallel port. To start the MCL-Net S24 application double-click on the MCL Net S24 V2 icon on the desktop or:

1. Click on the *Start* button in the Task Bar.
2. Select `Programs` group.
3. Select `MCL2` group.
4. Select `MCL-NetS24 V2`. The *MCL-NetS24* start-up window displays.



```
15:32:47.05 : NetS24 correctly started on IP : 157.235.23.78
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data0.dat for packets D0 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data1.dat for packets D1 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data2.dat for packets D2 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data3.dat for packets D3 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data4.dat for packets D4 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data5.dat for packets D5 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data6.dat for packets D6 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data7.dat for packets D7 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data8.dat for packets D8 and terminals 000
15:32:47.16 : Open file C:\MCL2\NETS24\DATA\data9.dat for packets D9 and terminals 000
15:32:47.27 : Start NetS24db DR -min
15:32:47.43 : Open pipe DR for packets DR and terminals 000
```

ID : 099 | IP : 157.235.23.78 | 15:58

Figure 1-1. MCL-Net S24 Start-Up Window

The window shows status information regarding the files for the terminals. The status bar displays the host logical ID address, host IP address and current time.

Stopping MCL-NetS24

The MCL-NetS24 program can be terminated by one of the following ways:

- select *Exit* in the *File* menu
- press **ALT-F4**
- press **Enter**
- select the `Exit` button.

Supervisor Window

The supervisor window is composed of three main parts:

- menu bar
- window displaying all messages received and sent to terminals
- status bar.

The window has a scroll bar which can only be accessed with the mouse. Only the last 1000 messages are stored in the memory of the window. To save more messages, set the trace options. Choose *Setup* in the *File* menu to open Setup window, and select the *Trace/Stats* tab.

The menu bar consists of:

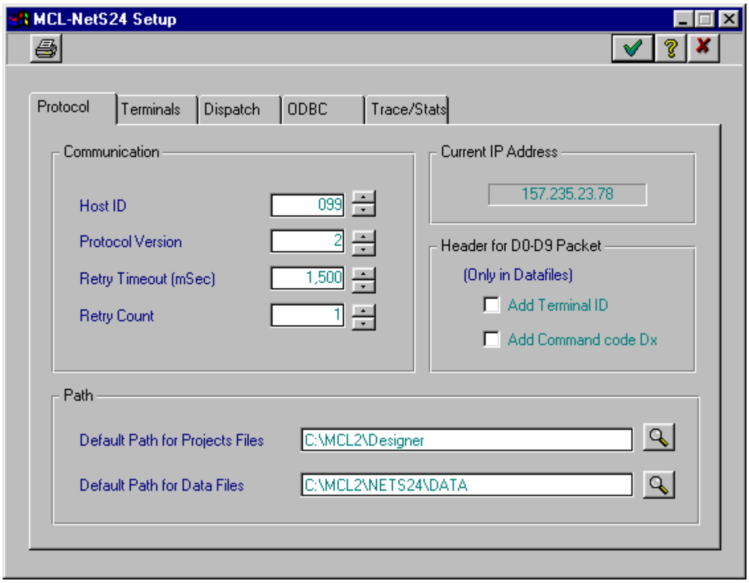
- File menu - used to setup or exit MCL-NetS24
- MCL-Net menu - checks the terminals configurations.
- Help menu - provides access to the on-line help file.

MCL-NetS24 Setup

The *Setup* option of the *File* menu accesses the Setup Window. It contains the Protocol, *Terminals*, *Dispatch*, *ODBC* and *Trace/Stats* tabs.

Protocol Tab

The Protocol tab accesses information about *Communication*, *Current IP Address*, *Header for D0...D9 packet*, and *Path*.



Communication

Host ID 099

Protocol Version 2

Retry Timeout (mSec) 1,500

Retry Count 1

Current IP Address

157.235.23.78

Header for D0-D9 Packet

(Only in Datafiles)

□ Add Terminal ID

□ Add Command code Dx

Path

Default Path for Projects Files C:\MCL2\Designer

Default Path for Data Files C:\MCL2\NETS24\DATA

Figure 1-2. Protocol Tab

Communication

The user can define the Host ID (range 001 – 254), the protocol version for file transfer, the retry time-out (range 100 - 3000) and the number of retries (range 1 – 3) concerning the data sent from MCL-NetS24 to the wireless units.

Header for D0..D9 packet

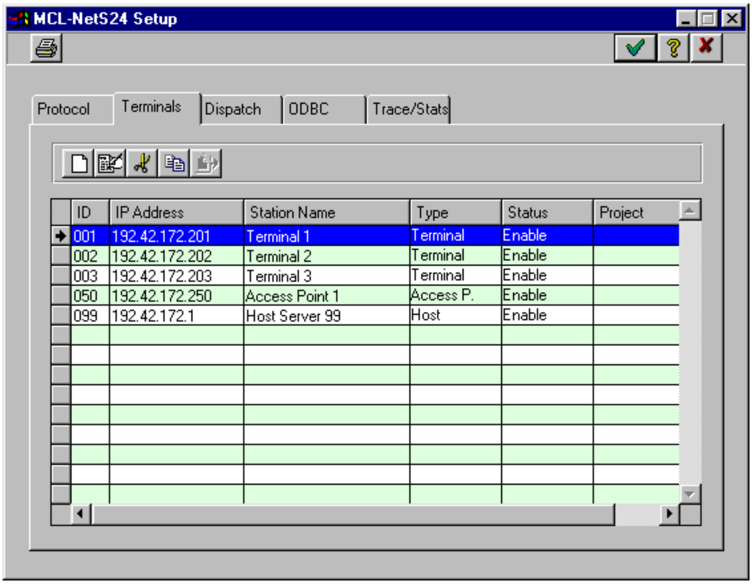
The user has the option to add the Terminal ID and/or the command, as a header, to each incoming data packet that is stored in data files.

Path

The default Path for Project files define the path of the directory where project files are stored. This path is necessary to find the correct SQL script file (PROJECT.MQD) related to a project. The default Path for Data Files is the path of the directory where data files are stored.

Terminals Tab

The Terminals tab accesses information about each individual terminal.



ID	IP Address	Station Name	Type	Status	Project
001	192.42.172.201	Terminal 1	Terminal	Enable	
002	192.42.172.202	Terminal 2	Terminal	Enable	
003	192.42.172.203	Terminal 3	Terminal	Enable	
050	192.42.172.250	Access Point 1	Access P.	Enable	
099	192.42.172.1	Host Server 99	Host	Enable	

Figure 1-3. Terminals Tab

This window lists the addresses of all the RF devices connected to the MCL-NetS24 communication program. This list allows MCL-NetS24 to make the correspondence between the Terminal ID and the IP address of each RF device.

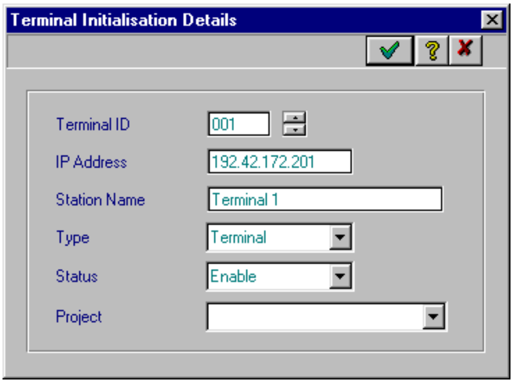
The CHECK process also uses this list. Access points and hosts can also be defined in the list so that the CHECK process includes them.

This Station ID Configuration list is stored in the TERMINAL.DAT file.

The Terminal ID must be unique in the system and the range is from 001 to 254. The IP Address must also be unique and the range depends on your Network Class.

Note: *The ID and IP address values also must be defined on the terminal itself.*

Double-clicking on a terminal line allows you to define the Terminal ID, the IP address, the station name, the device type (Terminal, Access point or Host), the device status (enable or disable) and the default project assigned to the terminal (optional parameter).


Terminal Initialisation Details

Terminal ID 001

IP Address 192.42.172.201

Station Name Terminal 1

Type Terminal

Status Enable

Project

Figure 1-4. Terminal Initialization Screen

Dispatcher Tab

The data dispatcher is a module that enables incoming data packets to be “routed” to the right host application or file, depending on Terminal ID (sender) and Packet Type. For instance, all type “D0” packets from terminal number 001 and 002 can be routed to an inventory program (via a pipe) while all type “D1” packets from all terminals are routed to the receiving good program (via another pipe).

The Data Dispatcher allows you to perform a “Load Balance” by giving the possibility to assign multiple pipes in parallel for the same criteria (Terminal ID and Packet Type), each one attached to a copy of the same application. This parallel implementation gives better response when several terminals perform the same transaction at the same time.

The Data Dispatcher will then route each incoming packet to the first free pipe corresponding to the criteria (Terminal ID and Packet Type) or to the pipe containing the lowest number of pending transactions. Up to 128 identical pipes can be defined on one server. To each pipe defined in the Data Dispatcher, an application file (EXE) can be attached and then launched automatically at MCL-NetS24 startup. To be able to read and write data from/to the pipe, the host application can use the corresponding functions given in the MCLNetS24 DLL library (see *How To Interface DLL* on page 2-14).

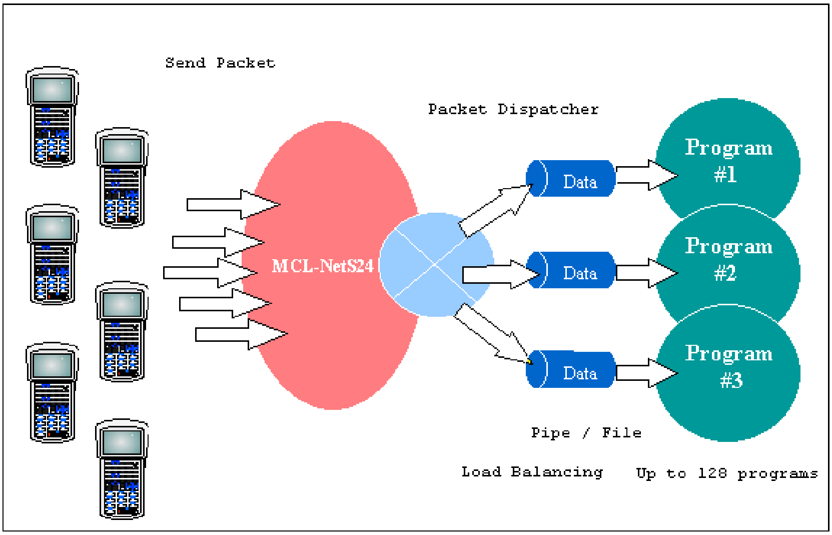
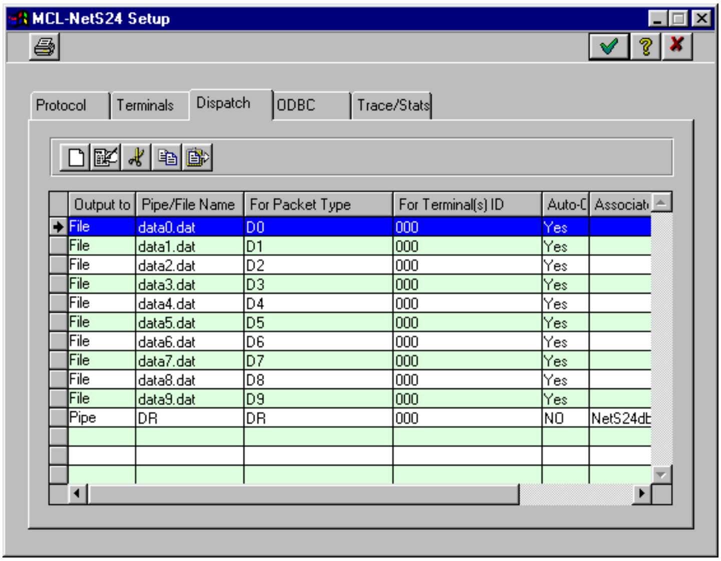


Figure 1-5. Data Dispatcher Illustration

The Data Dispatcher allows you to route incoming packets to text type data files. Each incoming packet will then be appended into the corresponding file, the host application can then, at any time, read the content of this file or rename the file for further operation without file sharing problems. The Dispatcher will create the file if it doesn't exist. This technique is usually applied when the terminals are only sending data to the host without any information coming back.

The Dispatcher tab sets up the data packet dispatcher to route the incoming packets to the corresponding application or file. The user can decide to dispatch selected incoming data (following packet type and terminal ID) to a file or to a named pipe that is attached to an application.



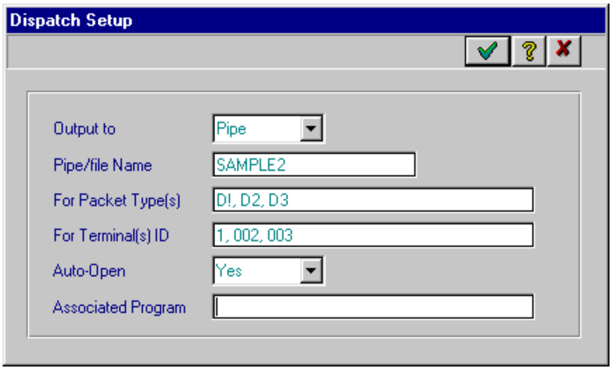
	Output to	Pipe/File Name	For Packet Type	For Terminal(s) ID	Auto-(	Associat
➔	File	data0.dat	D0	000	Yes	
	File	data1.dat	D1	000	Yes	
	File	data2.dat	D2	000	Yes	
	File	data3.dat	D3	000	Yes	
	File	data4.dat	D4	000	Yes	
	File	data5.dat	D5	000	Yes	
	File	data6.dat	D6	000	Yes	
	File	data7.dat	D7	000	Yes	
	File	data8.dat	D8	000	Yes	
	File	data9.dat	D9	000	Yes	
	Pipe	DR	DR	000	NO	NetS24db

Figure 1-6. Dispatcher Tab

The Auto-Open function, if set to YES, indicates that the MCL-NetS24 automatically creates the specified pipe or file at startup. If it set to no, the application program that requires the use of the specified data must create this pipe.

The user can define an executable file that is launched when MCL-NetS24 starts (Ex: NOTEPAD.EXE). The full path can be introduced (ex: C:\WINDOWS\NOTEPAD.EXE) or the path of a sub-directory of the MCL-NetS24 working directory (ex: \USERAPP\USERAPP.EXE).

Figure 1-7 illustrates that the data dispatcher creates and opens automatically a named pipe SAMPLE2 for packet types D1, D2, and D3 coming from terminals 001, 002, and 003.



Output to: Pipe

Pipe/file Name: SAMPLE2

For Packet Type(s): D!, D2, D3

For Terminal(s) ID: 1, 002, 003

Auto-Open: Yes

Associated Program:

Figure 1-7. Dispatch Setup

ODBC Tab

To access a SQL database through ODBC, the terminal sends a specific request ("DR" data packet) that contains the MQD filename, the SQL query number and the data to MCL-NetS24.

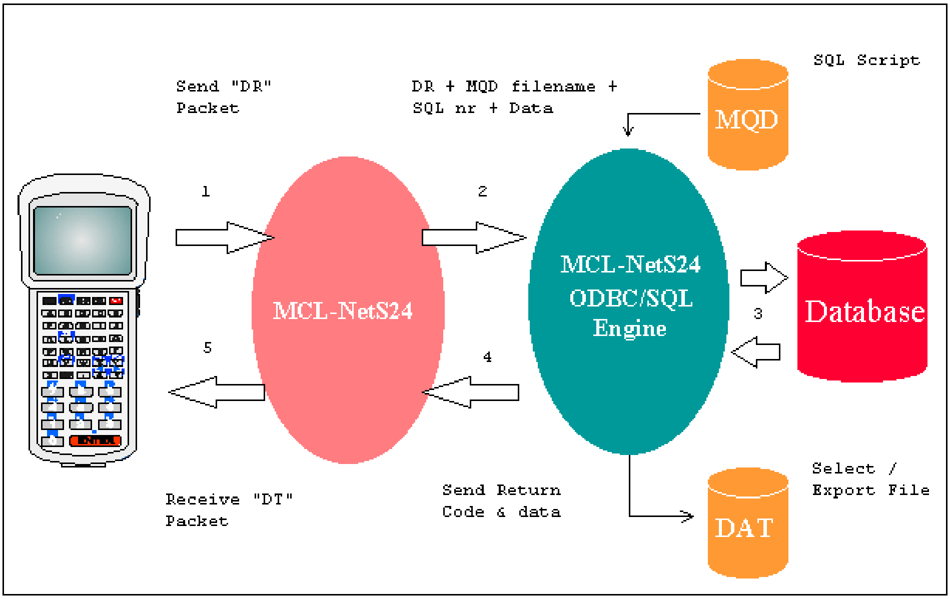
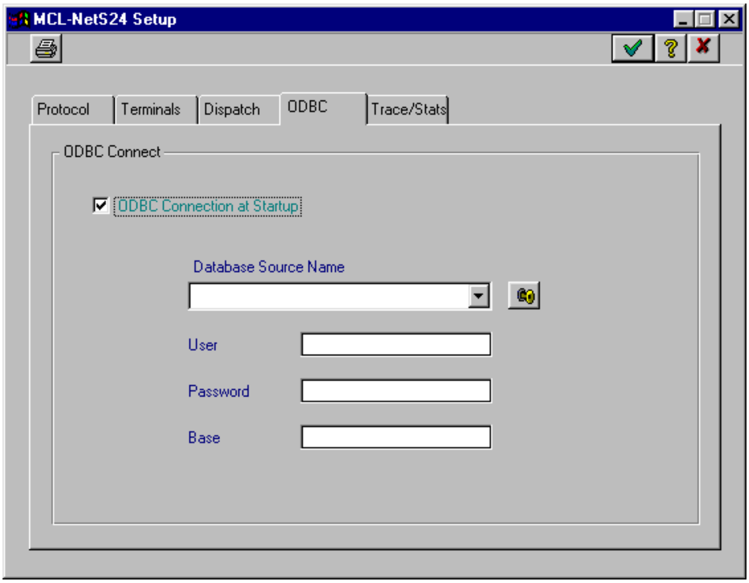


Figure 1-8. Accessing SQL Database

MCL-NetS24 transfer this request to the MCL-NetS24 OBDC/SQL engine that extracts the correct SQL query. This query comes from a query definition file named MQD. The MQD contains all queries generated by MCL-Designer that allows accessing data in a specified database for a specific project. The MQD file is stored in the specified project directory. This directory is defined in the protocol setup.

When the MCL-NetS24 ODBC engine finds (or cannot find) the corresponding data in the database, it sends return code and data to the terminal through the MCL-NetS24. The return code indicates the number of records that has found in the database.

The ODBC tab allows the user to select the database source name that has been defined previously in the Windows Control Panel – 32-bits ODBC settings.



☑ ODBC Connection at Startup

Figure 1-9. ODBC Tab

Selecting the ODBC Connection at Startup check-box automatically connects the MCL-NetS24 to the specified database. This window allows you to login to a database protected with username and password. Enter the real (not the logical) file name of the database in the Base field if necessary.

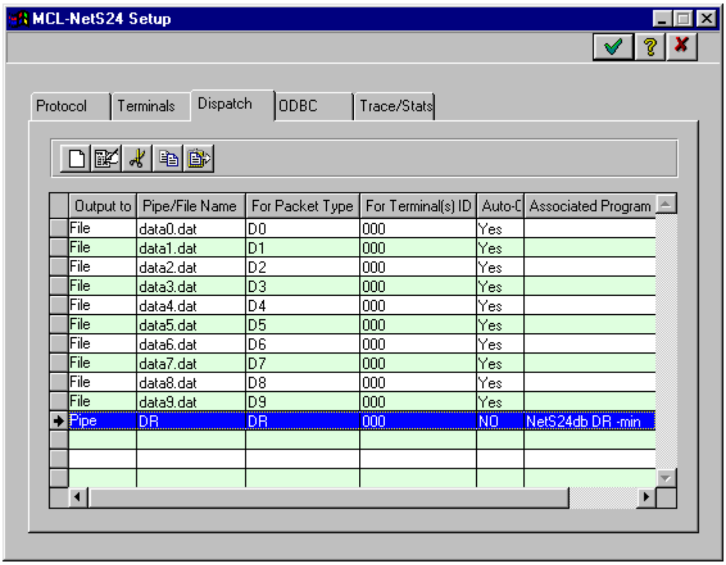
The MCL-NetS24 OBDC/SQL engine is an executable file named NETS24DB.EXE and is located in the working directory of the MCLNetS24. Its syntax is shown below:

NetS24db **pipename** *[-argument]*

where:

pipename: the name of the pipe that can be open automatically (Auto-Open set to Yes) by MCL-NetS24 dispatcher or by the user application. The Default pipename is DR and the default packet type is DR.

[-argument]: none: the NetS24db starts with normal window.
-min: the NetS24db starts minimized
-max: the NetS24db starts maximized

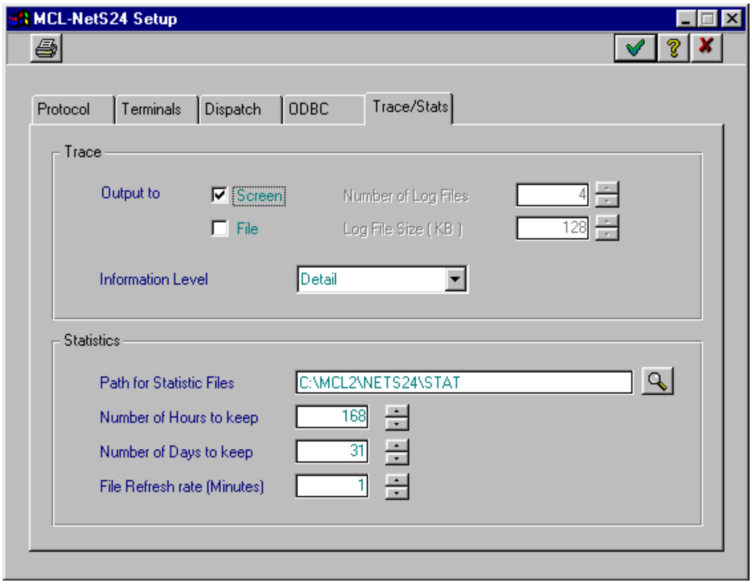


Output to	Pipe/File Name	For Packet Type	For Terminal(s) ID	Auto-C	Associated Program
File	data0.dat	D0	000	Yes	
File	data1.dat	D1	000	Yes	
File	data2.dat	D2	000	Yes	
File	data3.dat	D3	000	Yes	
File	data4.dat	D4	000	Yes	
File	data5.dat	D5	000	Yes	
File	data6.dat	D6	000	Yes	
File	data7.dat	D7	000	Yes	
File	data8.dat	D8	000	Yes	
File	data9.dat	D9	000	Yes	
Pipe	DR	DR	000	NO	NetS24db DR -min

Figure 1-10. Dispatch tab with Output to Pipe

Trace and Statistics

The Trace/Stats tab accesses information to define if tracing data is displayed and/or stored in log files.



Trace

Output to ☑ Screen

□ File

Number of Log Files 4

Log File Size (KB) 128

Information Level Detail

Statistics

Path for Statistic Files C:\MCL2\NETS24\STAT

Number of Hours to keep 168

Number of Days to keep 31

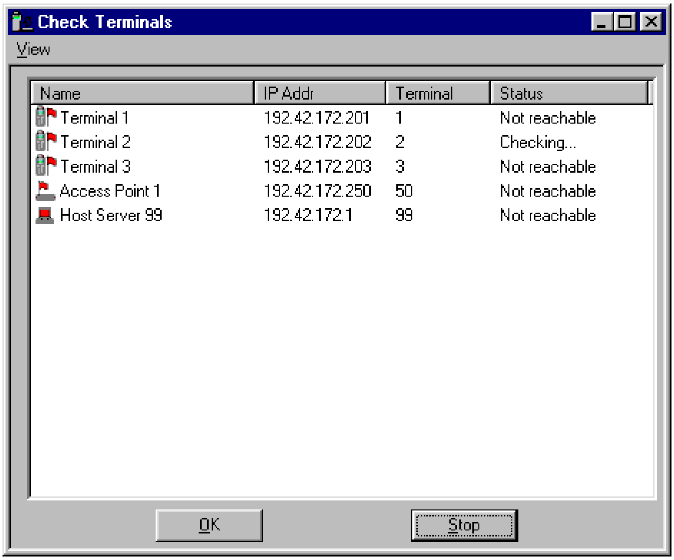
File Refresh rate (Minutes) 1

Figure 1-11. Trace and Stats Tab

The user is able to define the number, size, and level of information that is displayed or stored in log files. The Path for Statistic Files is the directory where MCL-Net S24 writes the statistic files. The user can define the number of log files created by MCL-NetS24 (range 1 – 999) and its size (range 1-999 k bytes). The user can also define the number of days (range 1 – 999) and hours (range 1 – 999) to keep the statistics data and the refresh rate of the statistics values.

Check Terminals

Choose the *Check Terminals* option under the *MCL-Net* menu to access the Check Terminals window. It displays information about each terminal's reachability.



Name	IP Addr	Terminal	Status
Terminal 1	192.42.172.201	1	Not reachable
Terminal 2	192.42.172.202	2	Checking...
Terminal 3	192.42.172.203	3	Not reachable
Access Point 1	192.42.172.250	50	Not reachable
Host Server 99	192.42.172.1	99	Not reachable

Figure 1-12. Check Terminals Tab

MCL-Net S24 refers to the information in TERMINAL.DAT to perform the communication verification (ping). Check terminals respond with one of three results:

Not reachable	the terminal is off or out of reach
Found	the terminal is present on the network
Connected	the terminal is connected to the host.

Click the Check button to start checking the terminals. Click the Stop button to halt terminal checking. If stopped, checking can be started again by clicking the Start button. The OK button exits the check window and returns to the supervisor.

TERMINAL.DAT File

Below is an example of the TERMINAL.DAT file:

```
[Base]
101,base RF Network ID

[DataFiles]

[List]
001,192.42.172.201,601,1,T,x,Terminal 1
002,192.42.172.202,601,1,T,x,Terminal 2
003,192.42.172.203,601,1,T,x,Terminal 3
050,192.42.172.250,601,1,A,x,Access Point 1
099,192.42.172.1,601,1,H,x,Host Server 99
```

Base and Data File sections are reserved for future extensions, but they are mandatory. Each line of the list section characterizes one terminal to be checked by MCL-Net S24. Each line is divided into 7 fields.

- The first field represents the logical address of the terminal (001-254).
- The second field is to the IP address of the terminal.
- The third field is the IP port application number (default = 601).
- The fourth and sixth fields are not used but mandatory.
- The fifth determines if the specified address corresponds to an RF terminal, an RF access point or a host.
- The seventh field is the name of the terminal. It may be modified, but an empty field is not allowed.

MCL-Net S24 Commands

MCL-NetS24 commands are subdivided into Terminal commands and Host commands. Host commands are sent from the host application to the terminal through the MCLNET24.DLL. The function used is MCLNet_SendData. To use these commands, the terminal receiving the data must be in Receive Mode.

FA =	File Append on terminal
FC =	File Copy on terminal
FD =	File Delete on terminal
FN =	File Rename on terminal
FR =	File Request to terminal
TF =	Transmit data file to terminal
TM =	Transmit MCL-Code program to the terminal

The Terminal commands come from the terminal and are executed on the host.

CR =	File Lookup
D0 to D9 =	Receive data packet
EX =	Execute an executable file
FA =	File append
FC =	File copy
FD =	File delete
FN =	File rename
FR =	File Request
SR =	Status request
TF =	Transmit data File
TR =	Time Synchronizing

Syntax of Commands

| = Separator. (ASCII 124)

xxx = Terminal identity (range: 001 to 254)

Terminal_File_Name = represented by a letter (from A to P) for data files and a digit (from 0 to 10) for an MCL program

Host_File_Name = may contain the directory and file extension (if no directory is specified, then the Default Path is used)

Terminal Commands

File Look-Up on Host (CR)

Function

Look up a file on the host.

Syntax

CR | *Host_File_Name* | *Key_to_search*

Example

RO | 099 | **CR** | PARTS.DAT | 1234567890123

Host_File_Name is a text file sorted on a key, and the key must be the first field in the file. If no directory is specified in the Host_File_Name, its location is defined in *Default Path Set-up*. MCL-Net S24 sends back to the terminal a **CT** command with data found in the record that matches the key.

CT | 0 =	Key not found
CT | 1 | Data_from_file =	Key found, data is the rest of the line. The first character after the key is skipped and not sent to the terminal
CT | 9 =	File not found

Receive Data Packet (Dx)

Function

Receives data packet (D0 - D9) from the terminal.

Syntax

RO | xxx | **Dx** | data

Example

RO | 099 | **D0** | 19971010 | 231022 | 1111

Dx = Data is appended into the data file which the path is defined in *Data Files Set-up*. The data file is created if it doesn't exist.

Execute an Executable File (EX)

Function

Runs an EXE program on the host.

Syntax

EX | [Drive:\Path\]Program name arg1 arg2 arg3…

Example

RO | 099 | **EX** | C:\5CWINDOWS\5CNOTEPAD.EXE MYDOC.TXT

Starts a Windows executable program in the specified directory. Otherwise it is in the same directory where MCL-Net S24 was installed.

File Append (FA)

Function

Appends a file on host.

Syntax

FA | File_1 | File_2

Example

RO | 099 | **FA** | ITEM.TXT | ITEM.BAK

ITEM.BAK is added to ITEM.TXT file. ITEM.BAK file remains unchanged.

File Copy (FC)

Function

Copies a file on the host.

Syntax

FC | New_file | Old_file

Example

RO | 099 | **FC** | ITEM.TXT | ITEM.BAK

The file ITEM.BAK is copied to the file ITEM.TXT. The two files are identical after the copy.

File Delete (FD)

Function

Deletes a file on host.

Syntax

FD | file_Name

Example

RO | 099 | **FD** | ITEM.TXT

The file ITEM.TXT is deleted on the host.

File Rename (FN)

Function

Renames a file on the host.

Syntax

FN | New_file | Old_file

Example

RO | 099 | **FN** | ITEM.TXT | ITEM.BAK

The file ITEM.BAK is renamed to ITEM.TXT.

File Request (FR)

Function

The terminal requests a data file from the host.

Syntax

FR | Host_File_Name | Terminal_File_Name

Example

RO | 099 | **FR** | C:\5CMCLD\5CPROJECT.PRJ\5CDATAFILE.DAT | A

The file DATAFILE.DAT located in C:\MCLD\PROJECT.PRJ is sent to the terminal.

Status Request (SR)

Function

Status Request to Host.

Syntax

SR | 1 = request MCL-Net S24 status from Host

SR | 3 | Host_File_Name = request file status from Host

Example

RO | 099 | **SR** | 3 | C:\5CDATA\5CITEM.DAT

MCL-Net S24 sends back to the terminal a **ST** command (status transmit) with the status of MCL-Net S24 or of the selected file.

<u>The MCL-Net S24 Status</u>

ST | 1 | MCL-Net S24 (TM) V1.01 | 20/12/1997 | 19971110114707 | 00000000 | 1

From left to right:

ST =	command status transmit
1 =	type of status; 1 = program
MCL-Net S24 ™ V1.01 =	the filename and its version
20/12/1997 =	date version
19971110114707 =	the current date and time stamp
00000000 =	reserved
1 =	reserved

<u>File Status</u>

ST | 3 | C:\DATA\FILEA.DAT | 012650 | 32856 | 19971110114707

From left to right:

ST =	command status transmit
3 =	type of status = file
C:\DATA\FILEA.DAT =	the filename
012650 =	The file size
32856 =	the computed CRC16 on the entire file
19971110114707 =	the last modification time stamp

Transmit Data File (TF)

Function

Transmits a data file to the host.

Syntax

TF | Terminal_File_Name | Host_File_Name

Example

RO | 099 | **TF** | A | C:\5CMCLD\5CPROJECT.PRJ\5CDATAFILE.DAT

The file A located on the terminal is stored on the host as C:\MCLD\PROJECT.PRJ\DATAFILE.DAT

Time Synchronizing (TR)

Function

Request the host to perform time synchronizing.

Syntax

TR

Example

RO | 099 | **TR**

The host sends a **TT** command which updates the date and time on the terminal.

Host Commands

Host commands are commands that are sent from the host application to a terminal through the MCLNET24.DLL. The function used is MCLNet_SendData. To use these commands, the terminal receiving the data must be in Receive Mode.

Conventions

cmd the command argument in the MCLNet_SendData function.

buffer the buffer argument in the MCLNet_SendData function.

Transmit Data File to the Terminal (TF)

Cmd **TF**

Buffer Host_File_Name | Terminal_File_Name

Example

C:\DATA\DATAFILE.DAT | A

The MCL-Net S24 terminal sends the file C:\DATA\DATAFILE.DAT to the terminal.

Transmit MCL-Code Program to the Terminal (TM)

Cmd **TM**

Buffer Host_File_Name | Terminal_File_Name

Example

Host_File_Name = C:\PGM\MAIN.MCL

Terminal_File_Name = 0

File Append on Terminal (FA)

Cmd **FA**

Buffer Old_File_name | New_File_name

Example

ITEM.BAK | ITEM.TXT

ITEM.BAK is added to ITEM.TXT file. ITEM.BAK file remains unchanged.

File Copy on Terminal (FC)

Cmd	**FC**
Buffer	Old_File_name | New_File_name

Example

ITEM.BAK | ITEM.TXT

The file ITEM.BAK is copied to the file ITEM.TXT.The two files are identical after the copy.

File Delete on Terminal (FD)

Cmd	**FD**
Buffer	File_name

Example

A.DAT

The file A.DAT on the terminal is deleted and its associated index file(s) if exists.

File Rename on Terminal (FN)

Cmd	**FN**
Buffer	Old_File_name | New_File_name

Example

ITEM.BAK | ITEM.TXT

The file ITEM.BAK is renamed to ITEM.TXT.

Data File Request to the Terminal (FR)

Cmd	**FR**
Buffer	Terminal_File_Name | Host_File_Name

Example

A.DAT | C:\DATA\DATAFILE.DAT

The terminal file A.DAT is sent to the host and stored as directory C:\DATA\DATAFILE.DAT.

Installing TCP/IP Drivers on Host

If TCP-IP drivers are not installed on your host computer, they must be installed prior to using MCL-Net24. To install the TCP/IP drivers:

1. On the Windows 95 desktop, double-click on the *My Computer* icon.

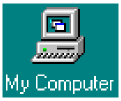


Figure 1-13. My Computer Icon

2. In the *My Computer* window, double-click on the *Control Panel* icon.

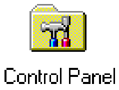


Figure 1-14. Control Panel Icon

3. In the *Control Panel* window, double-click on the *Network* icon.

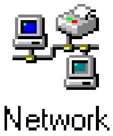
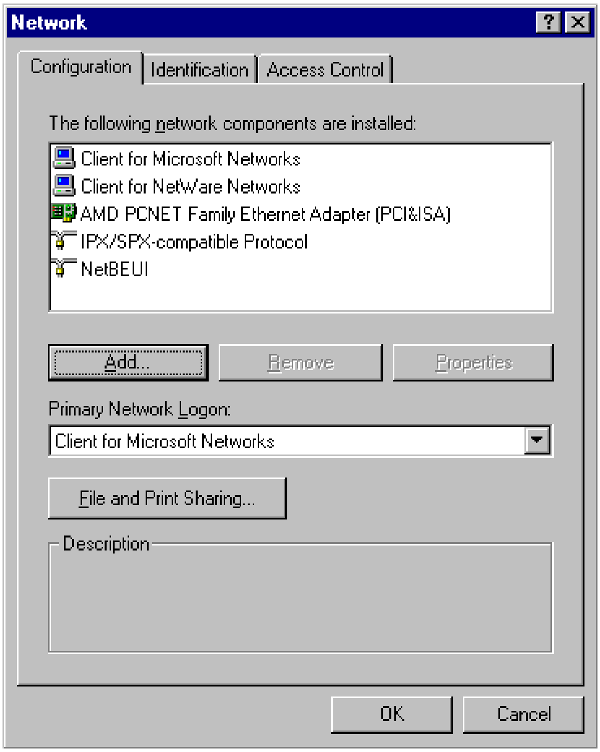


Figure 1-15. Network Icon

4. The *Network* window displays:



The following network components are installed:

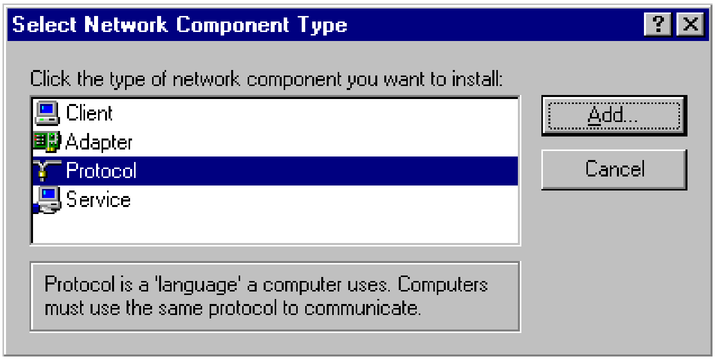
- Client for Microsoft Networks
- Client for NetWare Networks
- AMD PCNET Family Ethernet Adapter (PCI&ISA)
- IPX/SPX-compatible Protocol
- NetBEUI

Primary Network Logon: Client for Microsoft Networks

Figure 1-16. Network Window

5. Click on the Add button.

6. In the *Select Network Component Type* window, select `Protocol`.



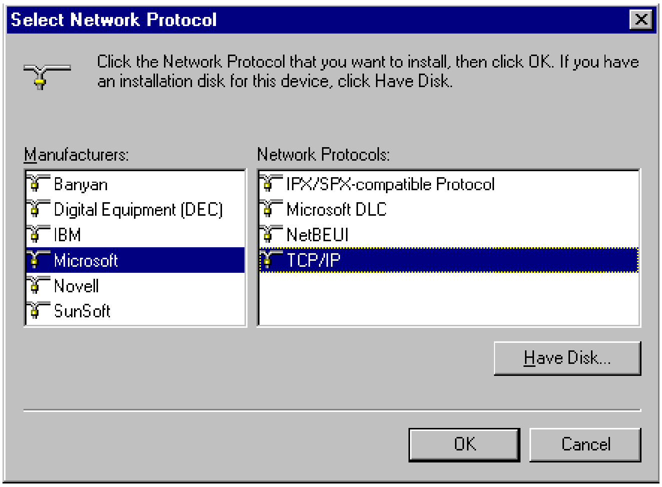
Select Network Component Type

Click the type of network component you want to install:

Protocol is a 'language' a computer uses. Computers must use the same protocol to communicate.

Figure 1-17. Select Network Component Type Window

7. Click on the `Add` button. The *Select Network Protocol* dialog box appears.



Select Network Protocol

Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this device, click Have Disk.

Figure 1-18. Select Network Protocol Dialog Box

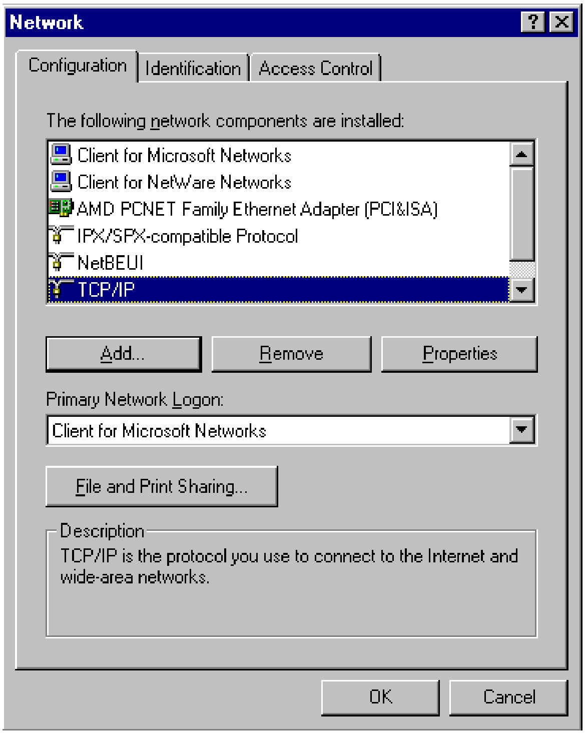
8. In the *Manufacturers* list box, select `Microsoft`.

9. In the *Network Protocols* list box, select TCP/IP.
10. Click on the OK button.
11. If prompted to reboot the computer, click on the YES button.

Define the Host IP Address

To define the host IP address:

1. On the Windows 95 desktop, double-click on the *My Computer* icon.
2. In the *My Computer* window, double-click on the *Control Panel* icon.
3. In the *Control Panel* window, double-click on the *Network* icon. The *Network* window displays.
4. In the *The following network components are installed* list box, select TCP/IP.
5. Click on the Properties button.



The following network components are installed:

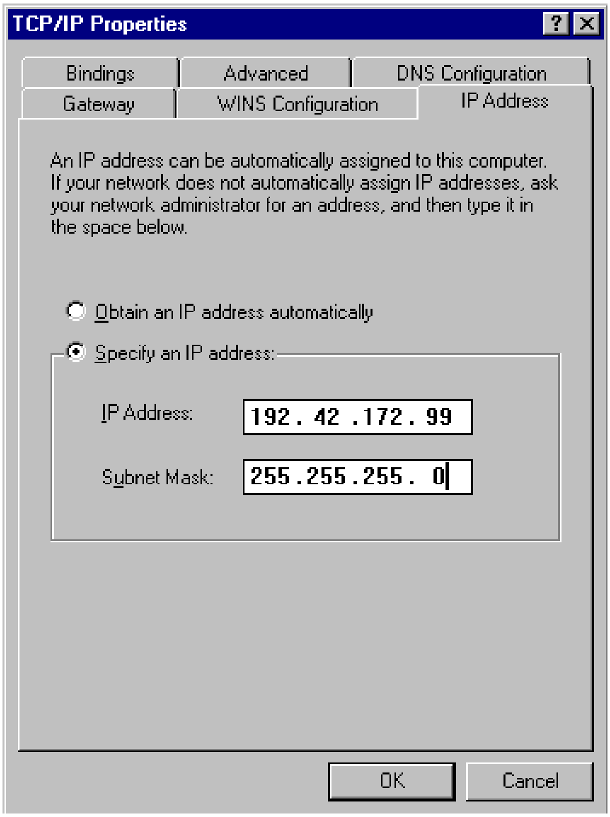
- Client for Microsoft Networks
- Client for NetWare Networks
- AMD PCNET Family Ethernet Adapter (PCI&ISA)
- IPX/SPX-compatible Protocol
- NetBEUI
- TCP/IP

Primary Network Logon: Client for Microsoft Networks

Description
TCP/IP is the protocol you use to connect to the Internet and wide-area networks.

Figure 1-19. Network Window - TCP/IP Properties

6. The *TCP/IP Properties* window appears.



An IP address can be automatically assigned to this computer. If your network does not automatically assign IP addresses, ask your network administrator for an address, and then type it in the space below.

○ Obtain an IP address automatically

● Specify an IP address:

IP Address: 192 . 42 .172 . 99

Subnet Mask: 255 .255 .255 . 0

Figure 1-20. TCP/IP Properties Window

7. Select the *IP Address* tab.
8. Click on *Specify an IP Address*:
9. Input the **IP Address** (example: 192.42.172.98).
10. Press the TAB key when finished.
11. Input the **Subnet Mask** (example: 255.255.255.0).
12. Click the OK button when finished.
13. When the message "You must restart your computer …" appears, click the YES button.

Installing MCL-Net S24 as a Windows NT Service

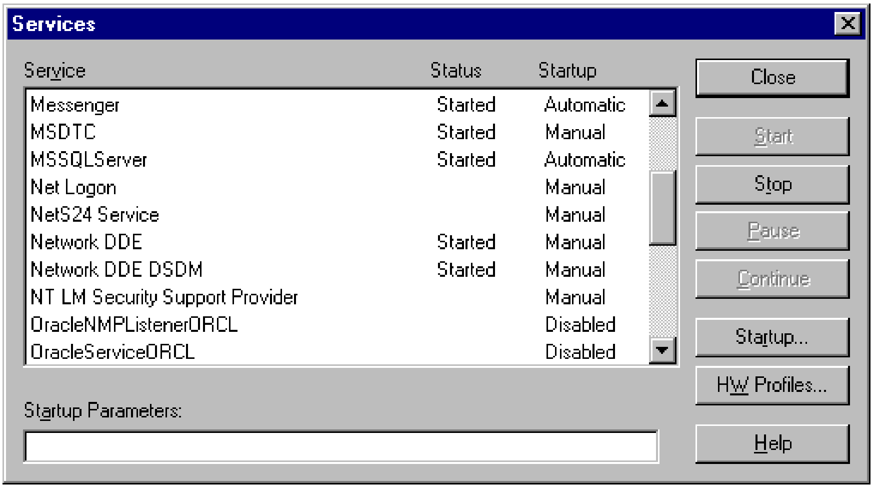
The executable file named `NetS24sv.exe` installs the MCL-NetS24 program as a Windows NT service. It is located in the working directory of the MCL-NetS24 (the default is C:\MCL2\NETS24). The `NetS24sv.exe` program is a 32 bits console application.

Have the administrator of the system install the MCL-NetS24 program first. It must be started in the normal mode to complete the necessary settings (terminal setup, dispatcher setup, etc.). When the settings are completed, exit the MCL-NetS24 program.

At this time, the user can start the NetS24sv program in a MS-DOS window with the argument `-install` to add the MCL-NetS24 in the services list.

```
NetS24SV –install
```

The NetS24 service appears in the list of service available on system (Control Panel – Services).



Services

Service	Status	Startup
Messenger	Started	Automatic
MSDTC	Started	Manual
MSSQLServer	Started	Automatic
Net Logon		Manual
NetS24 Service		Manual
Network DDE	Started	Manual
Network DDE DSDM	Started	Manual
NT LM Security Support Provider		Manual
OracleNMPListenerORCL		Disabled
OracleServiceORCL		Disabled

Figure 1-21. Services Window

The user can either modify the startup mode of the service (manual or automatic) or use the Windows NT standard command:

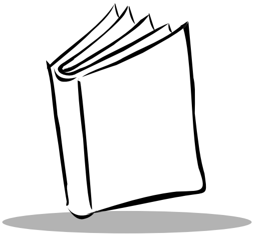
```
net start “nets24 service”
```

The MCL-NetS24 service may be stopped by the user using the process list (Control Panel – Services) or using the Windows NT standard command:

```
net stop “nets24 service”
```

To remove the service MCL-NetS24 from the services list, the user must stop the MCL-NetS24 service and start the NetS24sv with the option `–remove in a MS-DOS window.`

```
NetS24SV –remove
```

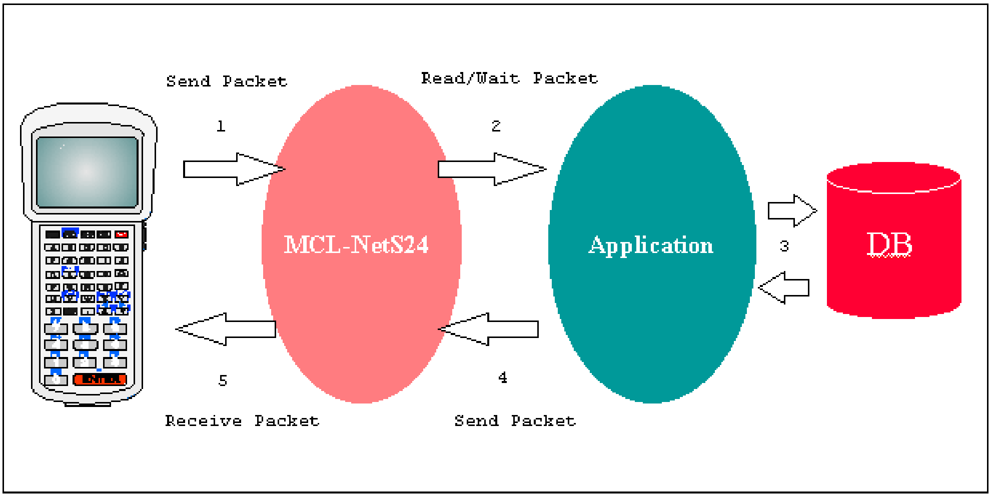


Chapter 2
MCL-Net S24 DLL

Introduction

The MCL-NetS24 provides a DLL interface to host applications in order to facilitate a client/ server relationship between the wireless devices and a host application. In addition to receiving operational terminal transaction data from the MCL-NetS24 and sending transaction-related data to a specific terminal via MCL-NetS24, the DLL aids network administration. The DLL enables to check if MCL-NetS24 is running; it also lets you start or stop MCL-NetS24 on your Windows NT / 95 / 98 environment.

Example: An application wants “D0” data packet type from terminal ID 001 and 002 and an other application wants “D1” data packet type from terminal ID 002 and 003. It’s the dispatcher job to route the right data packet to the right application via the correct named pipe.



In the above figure, one terminal sends data packets to one application via MCL-NetS24 and DLL. This application reads data from a database and sends it to the terminal to via the DLL and MCL-NetS24. On the terminal side, it is a send packet and receive packet transaction.

The MCL-Net S24 DLL (MCLNET24.DLL) is composed of ten functions:

- a function that checks if MCL-Net S24 is running.
- a function that checks if MCL-Net S24 is started, and if it is not, then starts it.
- a function that stops MCL-Net S24 if started.
- a function that checks if a terminal is connected.
- a function that waits for a connection of a terminal.
- a function that sends data to terminal.
- a function that receives data from terminal.
- a function that opens a named pipe.
- a function that closes a named pipe.
- a function that retrieves the IP address from the logical MCL ID.

Conventions

The following conventions are used:

- *int* and *long* represents a 32-bit signed integer (range -2^{31} to $2^{31}- 1$)
- *char* represents an 8 bit character (range 0 to 255).
- *int* * represents a near pointer to an array of 32 bit signed integers.
- *char* * represents a near pointer to an array of characters.

Return Code Standard Values

The return code is a 32-bit signed integer. A return code lower than 0 means that an error occurs. The ten significant error values are explained below:

-10	Error - Thread not found.
-9	Error - Too many processes/threads use the DLL (max 128).
-8	Error - Terminal not found in Terminal.dat.
-7	Error - Terminal not connected.
-6	Error - Terminal connected but not reachable.
-5	Error - A parameter is invalid.
-2	Error - System error.
-1	Error - Timeout.
0	Error - MCL-Net S24 not started.

MCLNet_Check

Function

Checks if MCL-Net S24 is running.

Syntax

```
int MCLNet_Check()
```

Return Value

A return code greater than 0 indicates that MCL-Net S24 is running.

DLL function ordinal number: 1

MCLNet_CheckAndStart

Function

Checks if MCL-Net S24 is running, if not, it starts MCL-Net S24 in foreground.

Syntax

int MCLNet_CheckAndStart(*int arg*)

where:

arg = interface mode:
0 = normal window
1 = maximized window
2 = minimized window

Return Value

A return code greater than 0 indicates that MCL-Net S24 is started correctly.

DLL function ordinal number: 2

Note: *The starting time of MCL-Net S24 must be less than 20 seconds.*

MCLNet_Stop

Function

Stop MCL-Net S24

Syntax

```
int MCLNet_Stop()
```

Return Value

A return code greater than 0 indicates that MCL-Net S24 is stopped correctly.

DLL function ordinal number: 3

MCLNet_CheckTerminal

Function

Check the state of a specific terminal.

Syntax

int MCLNet_CheckTerminal(*int Term*)

where:

Term = terminal number 1 to 254.

Return Value

A return code greater than 0 indicates that the terminal is connected and reachable.

DLL function ordinal number: 4

MCLNet_WaitTerminal

Function

Check terminal state and wait for a connection.

Syntax

int MCLNet_WaitTerminal(*int Term, long Timeout*)

where:

Timeout =	timeout in msec.
Term =	terminal number 1 to 254.

Return Value

A return code greater than 0 indicates that the terminal is connected and reachable.

DLL function ordinal number: 5

MCLNet_WaitData

Function

Receives data from MCL-Net S24.

Syntax

```
int MCLNet_WaitData (int * Term,
                     long Timeout,
                     char * Cmd,
                     char * Buffer,
                     int Maxlen)
```

where:

Timeout = timeout in msec.

Term = terminal number 1 to 254.

Cmd = expected command

Buffer = data input receive buffer

Maxlen = maximum length of input buffer

Return Value

A return code greater than 0 indicates that data has been received from the terminal which is connected and reachable. The returncode, if greater than 0, indicates the number of received characters.

DLL function ordinal number: 6

Notes

- Buffer which receives data must be declared to the effective maximum data length +1 because a 0 hexa is added as terminator.
- The variable Term must be initialized with a valid value (0 to 254).
- Received command (Packet Type) is copied in Cmd after the execution of the function.

MCLNet_SendData

Function

Sends data to a specified terminal.

Syntax

```
int MCLNet_SendData (int Term,
                     long Timeout,
                     char * Cmd,
                     char * buffer,
                     int Len)
```

where:

Timeout = timeout in msec.

Term = terminal number 1 to 254.

Cmd = command to send to the specified terminal (2 char.)

Buffer = data to send

Len = length of data

Return Value

A return code greater than 0 means that data has been sent correctly to the terminal which is connected and reachable.

DLL function ordinal number: 7

Note: *Timeout must be long enough in case of file or MCL-Code programs transfer to a specified terminal.*
If Timeout is set to 0, the function MCLNet_SendData is not waiting the acknowledgement from the terminal. This command exits immediately.

MCLNet_RetrieveIP

Function

Retrieve the terminal IP address from a specified logical MCL ID address. The correspondence is established in the TERMINAL.DAT file.

Syntax

int MCLNet_RetrieveIP (*int * Term*,
*char * szIP*,
*unsigned long * IP*)

where:

Term = terminal number 1 to 254.

szIP = IP address in string form (ex: 192.42.172.99).

IP = IP address in number form.

Return Value

A return code greater than 0 means that the IP address is found from terminal ID in TERMINAL.DAT file.

DLL function ordinal number: 8

MCLNet_OpenPipe

Function

Open a named pipe.

Syntax

int MCLNet_OpenPipe (*char * buffer*)

where:

Buffer = The name of the pipe (zero terminated string)

Return Value

A return code greater than 0 indicates that has no error to open the specified pipe.

DLL function ordinal number: 9

Note: *The name of the pipe must be defined in the dispatcher and MCL-NetS24 must be started.The name of the pipe may not exceed 60 bytes.*

MCLNet_ClosePipe

Function

Close a named pipe.

Syntax

int MCLNet_ClosePipe (*char * buffer*)

where:

Buffer = The name of the pipe (zero terminated string)

Return Value

A return code greater than 0 indicates that has no error to close the named pipe.

DLL function ordinal number: 10

Note: *The name of the pipe may not exceed 60 bytes.*

How To Interface DLL

With Visual Basic 5

Function Declaration

```
Private Declare Function NetStart Lib ".\MCLNET24.dll" Alias "MCLNet_CheckAndStart"
(ByVal WindowsType As Long) As Integer

Private Declare Function NetStop Lib ".\MCLNET24.dll" Alias "MCLNet_Stop" () As Integer

Private Declare Function NetOpenPipe Lib ".\MCLNET24.dll" Alias "MCLNet_OpenPipe"
(ByVal PipeName As String) As Integer

Private Declare Function NetClosePipe Lib ".\MCLNET24.dll" Alias "MCLNet_OpenPipe"
(ByVal PipeName As String) As Integer

Private Declare Function NetWait Lib ".\MCLNET24.dll" Alias "MCLNet_WaitData" _
(ByRef term As Long, ByVal TimeOut As Long, ByVal Cmd As String, ByVal Buffer As String,
ByVal MaxLen As Long) As Integer

Public Declare Function NetSend Lib ".\MCLNET24.dll" Alias "MCLNet_SendData" (ByVal term
As Long, ByVal TimeOut As Long, ByVal Cmd As String, ByVal Buffer As String, ByVal Length
As Long) As Integer
```

Variable Declaration

```
Public gPipeName As String
Dim ReturnCode As Integer
Dim term As Long
Dim SendFrame As String
Dim Buffer As String
Dim FrameSize As Long
Dim TimeOut As Long
```

Calling Function Examples

```
ReturnCode = NetStart(0)
ReturnCode = NetOpenPipe(PipeName)
ReturnCode = NetWait(term, 1000, gPipeName, Buffer, 512)
ReturnCode = NetSend(term, TimeOut, gPipeName, SendFrame, Len(SendFrame))
ReturnCode = NetClosePipe(PipeName)
ReturnCode = NetStop
```

With Visual C++ V4.x 32 bits

Function Declaration

```
__declspec(dllexport) int WINAPI MCLNet_CheckAndStart (int iStat);

__declspec(dllexport) int WINAPI MCLNet_Check ();

__declspec(dllexport) int WINAPI MCLNet_Stop ();

__declspec(dllexport) int WINAPI MCLNet_CheckTerminal (int iTerm);

__declspec(dllexport) int WINAPI MCLNet_WaitTerminal (int iTerm, long lTimeout);

__declspec(dllexport) int WINAPI MCLNet_WaitData (int * iTerm, long lTimeout, char *
szCmd, char * szBuffer, int iMaxlen);

__declspec(dllexport) int WINAPI MCLNet_SendData (int iTerm, long lTimeout, char * szCmd,
char * szBuffer, int iLen);

__declspec(dllexport) int WINAPI MCLNet_RetrieveIP (int iTerm, char * szIP, unsigned long
* pulIP);

__declspec(dllexport) int WINAPI MCLNet_OpenPipe (char * szBuffer);

__declspec(dllexport) int WINAPI MCLNet_ClosePipe (char * szBuffer);
```

Variable Declaration and Initialization

```
char    buffer[640];
int     iTerm=0;
long    lTimeout=250;
char    szCmd[64]="";
int     iMaxlen=sizeof(buffer);
```

Calling Function Examples

```
MCLNet_WaitData (&iTerm, lTimeout, &szCmd[0], &buffer[0], iMaxlen);

MCLNet_SendData(iTerm, 2000L, &szCmd[0], &buffer[0], strlen(buffer));

MCLNet_ClosePipe(m_saved_pipename.GetBuffer(m_saved_pipename.GetLength()));

MCLNet_OpenPipe(m_pipename.GetBuffer(m_pipename.GetLength()))
```

With Delphi 3.0

Function Declaration

```
function MCLNet_CheckAndStart (N: Integer): Integer stdcall; far; external
'.\MCLNet24.DLL' index 2;

function MCLNet_Stop: Integer stdcall; far; external '.\MCLNET24.DLL' index 3;

function MCLNet_CheckTerminal(pterm: Integer):Integer stdcall; far; external
'.\MCLNET24.DLL' index 4;

function MCLNet_WaitTerminal(pterm: Integer; pTimeout: Integer):Integer stdcall; far;
external '.\MCLNET24.DLL' index 5;

function MCLNet_SendData(pterm:Integer; pTimeout:Integer; Cmd:string; Buffer:string;
Len:Integer):Integer stdcall; far; external '.\MCLNET24.DLL' index 7;

function MCLNet_WaitData(var pterm:Integer; pTimeout:Integer; PipeName:pchar;
Buf:pchar; Len:Integer):Integer stdcall; far; external '.\MCLNET24.DLL' index 6;

function MCLNet_OpenPipe(pipe:string):Integer stdcall; far; external '.\MCLNET24.DLL'
index 9;

function MCLNet_ClosePipe(pipe:string):Integer stdcall; far; external '.\MCLNET24.DLL'
index 10;

Remarks: stdcall identify how to push arguments on the stack.
```

Variable Declaration

```
var
returncode:Integer;
Term: Integer;
Timeout: Integer;
StartType: Integer;
Buffer: array [0..512] of char;
RecCmd: array [0..512] of char;
Pipename: array[0..63] of char;
```

Variable Initialization

```
Term:= 1;
StartType:= 0;
Cmd:= 'D0';
Buffer:= '0123456789';
Timeout:= 20000;
Maxlen:= 10;
Pipename:=’SAMPLEPIPE’;
```

Calling Function Examples

```
returncode:= MCLNet_CheckAndStart(StartType);

returncode:= MCLNet_Stop();

returncode:= MCLNet_CheckTerminal(term);

returncode:= MCLNet_SendData(Term,Timeout,Cmd,Buffer,Maxlen);

returncode:= MCLNet_WaitTerminal(Term,Timeout);

returncode:= MCLNet_WaitData(Term,Timeout,@PipeName,@Buffer,255);

returncode:= MCLNet_OpenPipe(Pipename);

returncode:= MCLNet_ClosePipe(Pipename);
```

A Microsoft Visual Basic Version 5 Sample

Introduction

This sample program is written with Microsoft Visual Basic 5. Click on DLL.EXE icon to start MCL-Net S24 and wait for data coming from the terminal. The Exit button stops the MCL-Net S24 and returns to Windows.

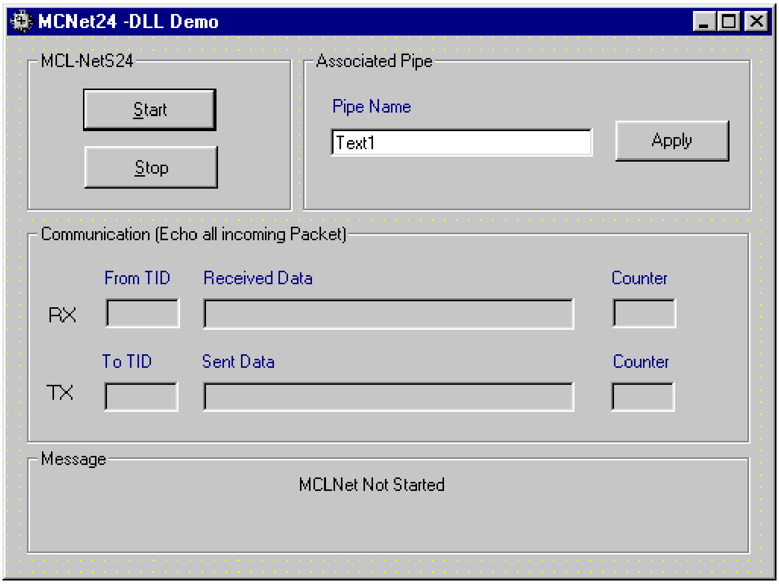
The user interface consists of six text boxes which contain terminal's identity, receive (or send) data, and record counter, two buttons that starts and stops MCLNet24 and an edit box dedicated for input the pipe name.

The main function of this sample is that selected incoming data are re-sent to the terminal via the DLL functions.

The terminal sends a "D0" command with some data (max 512 bytes). The sample program waits for a message "D0" that comes from MCL-Net S24 using the MCLNet_WaitData function inside the MCLNET24.DLL. When a "D0" message is received, the sample program re-sends the same data to the terminal using the MCLNet_SendData DLL function.

The first part of the listing is the MCL-Net S24 DLL functions and after the variables declaration.

User Interface



Text1

MCLNet Not Started

Figure 2-1. MCL-Net S24 DLL Demo Window

Counter indicates the number of received and sent data packets.

The message panel displays the error and warning messages.

Program Source

```
Option Explicit

'
'Declaration of DLL calls
'
'Call to Start MCL-NetS24
Private Declare Function NetStart Lib ".\MCLNET24.dll" Alias “MCLNet_CheckAndStart”
(ByVal WindowsType As Long) _
As Integer

'Call to Stop MCL-NetS24
Private Declare Function NetStop Lib ".\MCLNET24.dll" Alias "MCLNet_Stop" () As Integer

Private Declare Function NetOpenPipe Lib ".\MCLNET24.dll" Alias "MCLNet_OpenPipe"
(ByVal PipeName As String) As Integer

Private Declare Function NetClosePipe Lib ".\MCLNET24.dll" Alias "MCLNet_OpenPipe"
(ByVal PipeName As String) As Integer

'Call to Get Data from MCL-NetS24
Private Declare Function NetWait Lib ".\MCLNET24.dll" Alias "MCLNet_WaitData" _
(ByRef term As Long, ByVal TimeOut As Long, ByVal Cmd As String, ByVal Buffer As String,
ByVal MaxLen As Long) As Integer

'Call to Send Data to MCL-NetS24
Public Declare Function NetSend Lib ".\MCLNET24.dll" Alias "MCLNet_SendData" (ByVal term
As Long, ByVal TimeOut As Long, ByVal Cmd As String, ByVal Buffer As String, ByVal Length
As Long) As Integer

'Global Variable
Public gPipeName As String


Public Function StartNet24(Interface As Object) As Boolean
'Start MCL-NetS24 and return True if successful
Dim ReturnCode As Integer
 'Start MCLNET
 ReturnCode = NetStart(0)
 ' MclNet successfully started returns a positive code
 If ReturnCode <= 0 Then
  'Translate error status!
  With Interface.lblStatus
   .Caption = "Unable to start MCLNet " & "Message Code: " & GetCode(ReturnCode)
   .ForeColor = vbRed
  End With
  'Return Ok
  StartNet24 = False

 Else

  'Successful MCL-NetS24 start!
  With Interface.lblStatus
   .Caption = "MCLNet24 successfully started"
   .ForeColor = vbBlue
  End With
  'Open Named pipe between Mcl-NetS24 Dll and Mcl-NSet24 program
  If gPipeName <> "" Then
   NetOpenPipe gPipeName
  End If

  'Return Bad
  StartNet24 = True

 End If
End Function

Public Sub StopNet()
'Stop MclNet24 and close the previously open Pipe
Dim Wait1sec As Date

  NetClosePipe gPipeName
  NetStop

End Sub

Public Function OpenPipeNet(PipeName As String, Interface As Object)
'Open the specific named pipe
Dim ReturnCode As Integer
  ReturnCode = NetOpenPipe(PipeName)
 Select Case ReturnCode
 Case Is > 0
  'Transaction is Ok
  OpenPipeNet = ReturnCode

 Case -1
  'Time Out
  OpenPipeNet = 0

 Case Else
  'Get error code
  Interface.lblStatus = "Open Pipe Error " & "Message Code: " & GetCode(ReturnCode)
  Interface.lblStatus.FontSize = 10
  Interface.lblStatus.ForeColor = vbRed
  'Return Error Condition
  OpenPipeNet = -1

 End Select

End Function

Public Function ClosePipeNet(PipeName As String, Interface As Object)
'Close the previously open Pipe
Dim ReturnCode As Integer
  ReturnCode = NetClosePipe(PipeName)
 Select Case ReturnCode
 Case Is > 0
  'Transaction is Ok
  ClosePipeNet = ReturnCode

 Case -1
  'Time Out
  ClosePipeNet = 0

 Case Else
  'Get error code
  Interface.lblStatus = "Close Pipe Error " & "Message Code: " & GetCode(ReturnCode)
  Interface.lblStatus.FontSize = 10
  Interface.lblStatus.ForeColor = vbRed

  'Return Error Condition
  ClosePipeNet = -1

 End Select

End Function


Public Function WaitData(Interface As Object) As Integer
'Wait Data from the terminal through the opened pipe and NetS24
Dim ReturnCode As Integer
Dim Buffer As String
Dim term As Long

 'Initialize Buffer
 Buffer = String(512, vbNullChar)


 'Argument received :Term as Terminal Id (001 to 254), Receive data (Buffer)
 'Arguments supplied: Time Out(1000 mSec), Pipe Name (gPipeName), Receive Packet size
(512 Bytes)
 ReturnCode = NetWait(term, 1000, gPipeName, Buffer, 512)
 'Check Return Code
 Select Case ReturnCode
 Case Is > 0
  'Transaction is Ok
  Interface.txtIn = Buffer
  Interface.txtTIDIn.Text = Format$(term, "000")
  Interface.txtCountIn = Str(Val(Interface.txtCountIn) + 1)
  Interface.tmrLoop.Interval = 300
  SendData Interface
  WaitData = ReturnCode

 Case -1
  'Time Out
  WaitData = 0

 Case Else
  'Get error code
  Interface.lblStatus = "Input Error " & "Message Code: " & GetCode(ReturnCode)
  Interface.lblStatus.FontSize = 10
  Interface.lblStatus.ForeColor = vbRed
  WaitData = -1

 End Select
End Function

Private Sub SendData(Interface As Object)

'Return Echo of the received data

Dim ReturnCode As Integer
Dim term As Long
Dim SendFrame As String
Dim Buffer As String
Dim FrameSize As Long
Dim TimeOut As Long

 'Send Data to MclNet24
 SendFrame = Interface.txtIn.Text
 term = CLng(Interface.txtTIDIn)
 TimeOut = 5000

 ReturnCode = NetSend(term, TimeOut, gPipeName, SendFrame, Len(SendFrame))

 'Check Return Code
 Select Case ReturnCode
 Case Is > 0
  'Transaction is Ok
  Interface.txtIn = SendFrame
  Interface.txtTIDOut.Text = Format$(term, "000")
  Interface.tmrLoop.Interval = 300
  Interface.txtCountOut = Str(Val(Interface.txtCountOut) + 1)
  Interface.txtOut = Interface.txtIn

 Case -1
  'Time Out

 Case Else
  'Get error code
  Interface.lblStatus = "Output Error " & "Message Code: " & GetCode(ReturnCode)
  Interface.lblStatus.FontSize = 10
  Interface.lblStatus.ForeColor = vbRed

 End Select

End Sub

Private Function GetCode(Code As Integer) As String
'This function translates MCLNet24 numeric error code in
' their corresponding error text

   Select Case Code
   Case 0: GetCode = "MCL-NetS24 not Started"
   Case -1: GetCode = "Time Out"
   Case -2: GetCode = "System Error"
   Case -3: GetCode = "Registry information not found"
   Case -4: GetCode = "Cannot allocate shared memory"
   Case -5: GetCode = "Bad parameter"
   Case -6: GetCode = "Terminal connected but not reachable"
   Case -7: GetCode = "Terminal not connected"
   Case -8: GetCode = "Terminal not Defined"
   Case -9: GetCode = "Too many DLL users (max 128 threads)"
   Case -10: GetCode = "Thread not found"
   Case Is < -10: GetCode = "Error " + Str(Code)
   End Select

End Function

'
'This program Open a named pipe and Echo the received packet(s)
'
Option Explicit

Private Sub cmdPipeOpen_Click()
'Change pipe name
Dim DummyCode As Integer 'Return Code for Pipe opening not used here
```

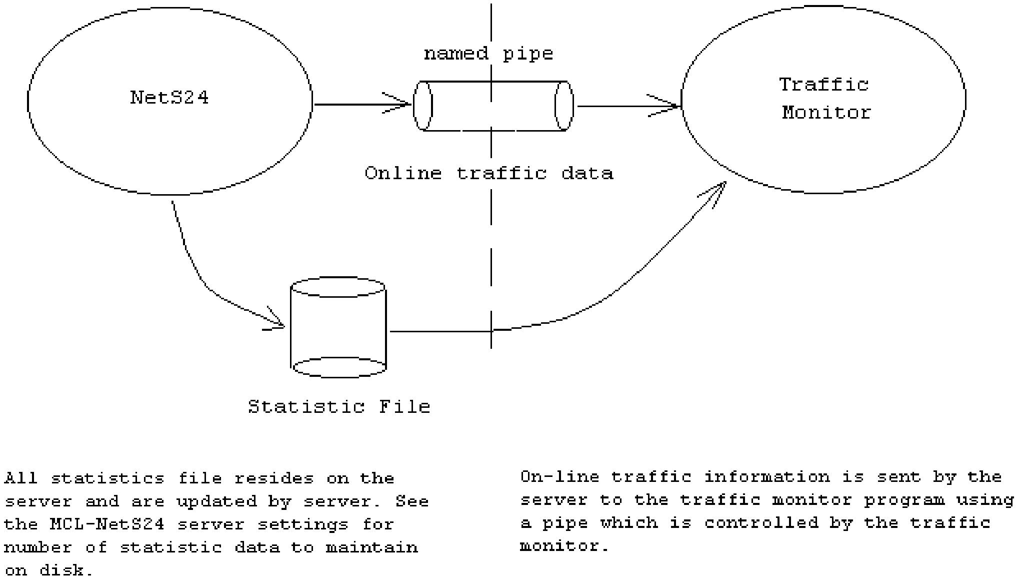
```
'Check if a Pipe name is provided by the user
      If txtPipeName <> "" Then
   'If Pipe name different then close previous pipe before opening a new one
         If gPipeName <> txtPipeName Then

            'Check if closing pipe is successful
            If ClosePipeNet(gPipeName, Me) > 0 Then

            gPipeName = txtPipeName
            DummyCode = OpenPipeNet(gPipeName, Me)

            End If

         End If

      Else

      'If no Pipe name then display a warning
      MsgBox ("Please provide a PipeName")
      txtPipeName.SetFocus

      End If
End Sub

Private Sub cmdStart_Click()
'Start MCL-NetS24

      'Check if starting mCL-nets24 is successful
      ' and provide the calling object reference ('frmDLL' form)
      If StartNet24(Me) Then

            'If successfully started initialize looping timer
            tmrLoop.Enabled = True

            'Toggle between Start and Stop button to avoid
            ' multiple start and stop
            cmdStop.Enabled = True
            cmdStart.Enabled = False

      End If
 End Sub

Private Sub cmdStop_Click()
'Stop Mclnet24
      StopNet

      'Toggle between Start and Stop button to avoid
      ' multiple start and stop
      cmdStart.Enabled = True
      cmdStop.Enabled = False

End Sub

Private Sub Form_Load()
'Set default pipe name
   txtPipeName = "D0"
   gPipeName = "D0"
End Sub

Private Sub Form_Unload(Cancel As Integer)
'Unload the VB DLL demo
         Unload Me
End Sub

Private Sub tmrLoop_Timer()
'Timer triggering loop while waiting data
'Stop when an error is encountered

   'On error from NetS24 exit the timer
   If WaitData(Me) = -1 Then Exit Sub

      'Reset the Timer
  tmrLoop.Interval = 300
End Sub
```

Chapter 3
MCL-Net S24 Traffic Monitor

Introduction

MCL-NetS24 Monitor is a 32-bit Windows 95/98/NT/Radio traffic monitor. It provides the ability to monitor radio communication transactions and statistics between the MCL-NetS24 server and mobile terminals using Spectrum24 radio cards and MCL applications.

MCL-NetS24 traffic monitor can be installed locally (on the MCL-NetS24 server) or remotely (on a remote workstation). For remote operation, the server must be running Windows NT and a TCP/IP connection must be available between the remote and the server computer.



All statistics file resides on the server and are updated by server. See the MCL-NetS24 server settings for number of statistic data to maintain on disk.

On-line traffic information is sent by the server to the traffic monitor program using a pipe which is controlled by the traffic monitor.

Figure 3-1. Traffic Monitor Map

User Interface

MCL-NetS24 Traffic monitor allows to the user to:

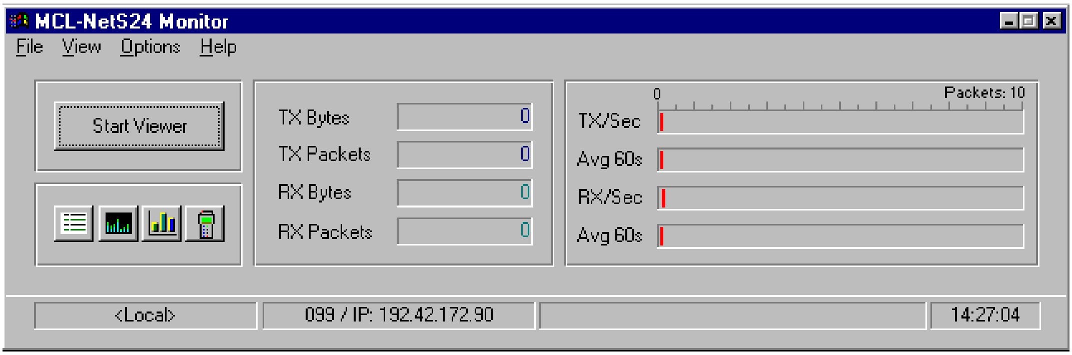
- Start and stop MCL-NetS24 communication server (if not installed as a Windows NT service).
- Monitor on-line communication transactions.
- Compute and display miscellaneous traffic statistics.
- Print statistics and graphs.

Starting MCL-NetS24 Traffic Monitor

To start the MCL-Net S24 Traffic monitor application double-click on the MCL Net S24 Traffic Monitor icon on the desktop or:

1. Click on the *Start* button in the Task Bar.
2. Select `Programs` group.
3. Select `MCL2` group.

4. Select `MCL-NetS24 Traffic Monitor`. The *MCL-Net S24 Traffic Monitor* start-up window displays.



TX Bytes 0
TX Packets 0
RX Bytes 0
RX Packets 0

<Local>
099 / IP: 192.42.172.90
14:27:04

Figure 3-2. MCL-Net S24 Traffic Monitor

The window displays the Start/Stop Viewer button and four display buttons.

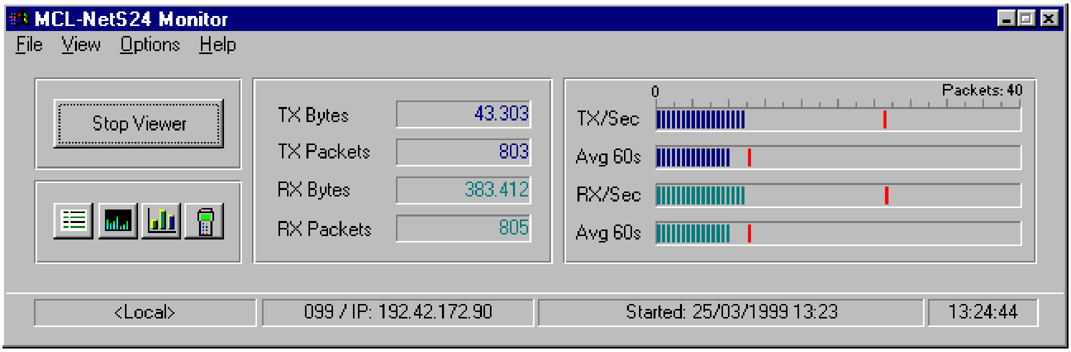
Online Data Display on page 3-6.

Online Graph Display on page 3-7.

Statistics Graph Display on page 3-8.

Terminals Statistics Graph Display on page 3-9.

To start the Traffic monitor, click on the *Start Viewer* button.



TX Bytes 43.303
TX Packets 803
RX Bytes 383.412
RX Packets 805

<Local>
099 / IP: 192.42.172.90
Started: 25/03/1999 13:23
13:24:44

Figure 3-3. MCL-Net S24 Traffic Monitor

To specify the units for the gauge scales, choose *Gauge* from the *Options* menu.

Tx/Sec Number of bytes (or packets) sent in one second to the terminals.

Tx Bytes Total number of bytes sent to mobile units since Traffic monitor started.

Tx Packets Total number of packets sent to mobile units since Traffic monitor started.

Avg 60s Average number of bytes (or packets) sent in one second (average computed on the last 60 seconds).

Rx/Sec Number of bytes (or packets) received in one second from the terminals.

Rx Bytes Total number of bytes received from mobile units since Traffic monitor started.

Rx Packets Total number of packets received from mobile units since Traffic monitor started.

Avg 60s Average number of bytes (or packets) received in one second (average computed on the last 60 seconds).

Peak value is maintained by the red line and the scale is automatically adjusted.

File Menu

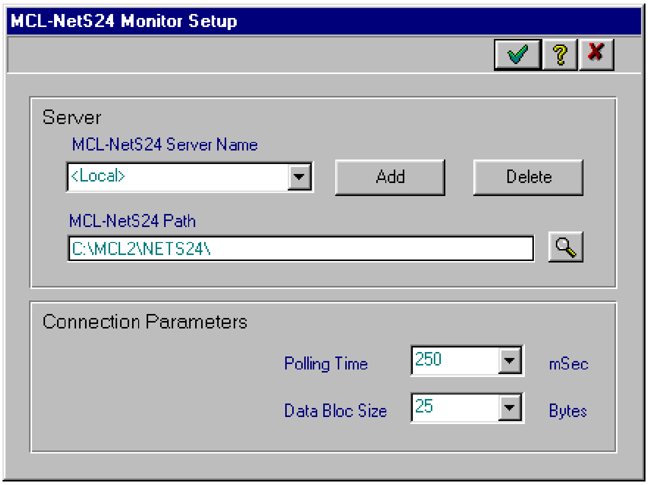
The user can start or stop the MCL-NetS24 communication server, set up the MCL-NetS24 traffic monitor, set up the MCL-NetS24 communication server, and print the online and statistics graphs using the File menu option.

MCL-NetS24 Setup

When *MCL-NetS24 Setup...* is chosen from the *File* menu, the MCL-NetS24 Setup window appears. Refer to *MCL-NetS24 Setup* on page 1-4 for more information.

MCL-NetS24 Traffic Monitor Setup

When *MCL-NetS24 Monitor Setup...* is chosen from the *File* menu, the MCL-NetS24 Monitor Setup dialog box appears:



Server

MCL-NetS24 Server Name <Local>

MCL-NetS24 Path: C:\MCL2\NETS24\

Connection Parameters

Polling Time: 250 mSec

Data Bloc Size: 25 Bytes

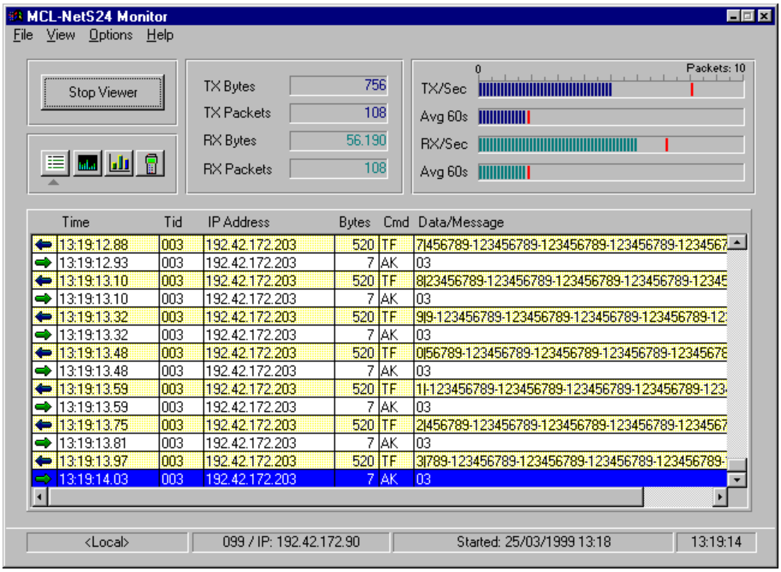
Figure 3-4. MCL-NetS24 Traffic Monitor Setup Dialog Box

It allows the user to specify:

- The MCL-NetS24 server communication location. The **MCL-NetS24 Server Name** can be <*Local*> if it is installed on the same PC or the server name if it is installed remotely. The **MCL-NetS24 Path** specifies the working directory of the communication server.
- The connection parameters. The **Polling Time** defines the delay between each interrogation to the MCL-NetS24 server. The **Data Bloc Size** defines the maximum length of user data to receive from the server. The data coming from the MCL-NetS24 server is truncated if it is longer than the Data Bloc Size value. This technique minimizes the number of bytes sent by the server to the traffic monitor (useful for remote traffic monitoring using low speed communication line). This has no effect on the displayed values and statistics. For minimum data traffic, set polling time to maximum value and data bloc size to minimum (ex: 1000 msec/10 bytes).

Online Data Display

Click the *Online Data Display* button to expand the window and display the online data. All communication frames between the server and the terminals are displayed in this window.



TX Bytes	756
TX Packets	108
RX Bytes	56.190
RX Packets	108

	Time	Tid	IP Address	Bytes	Cmd	Data/Message
←	13:19:12.88	003	192.42.172.203	520	TF	7|456789-123456789-123456789-123456789-1234567
→	13:19:12.93	003	192.42.172.203	7	AK	03
←	13:19:13.10	003	192.42.172.203	520	TF	8|23456789-123456789-123456789-123456789-12345
→	13:19:13.10	003	192.42.172.203	7	AK	03
←	13:19:13.32	003	192.42.172.203	520	TF	9|9-123456789-123456789-123456789-123456789-12
→	13:19:13.32	003	192.42.172.203	7	AK	03
←	13:19:13.48	003	192.42.172.203	520	TF	0|56789-123456789-123456789-123456789-1234567
→	13:19:13.48	003	192.42.172.203	7	AK	03
←	13:19:13.59	003	192.42.172.203	520	TF	1|-123456789-123456789-123456789-123456789-123
→	13:19:13.59	003	192.42.172.203	7	AK	03
←	13:19:13.75	003	192.42.172.203	520	TF	2|456789-123456789-123456789-123456789-1234567
→	13:19:13.81	003	192.42.172.203	7	AK	03
←	13:19:13.97	003	192.42.172.203	520	TF	3|789-123456789-123456789-123456789-123456789-
→	13:19:14.03	003	192.42.172.203	7	AK	03

<Local>

099 / IP: 192.42.172.90

Started: 25/03/1999 13:18

13:19:14

Figure 3-5. Online Data Display

← : Traffic from server to terminals (TX).

→ : Traffic from terminals to server (RX).

Time: Displays the exact time (in 24 hour-mode) the transaction occurred.

Tid: Represents the logical address of the Spectrum24 terminal.

IP Address: Displays the address of the local computer.

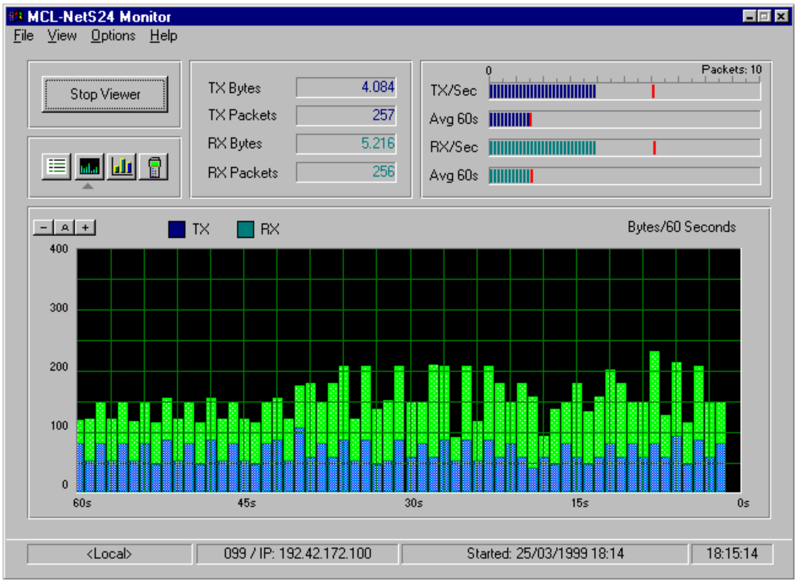
Bytes: Displays the effective data length.

Cmd: Displays the sent or received command.

Data/Message: Displays the data sent or any error message from the terminal.

Online Graph Display

Click the *Online Graph Display* button to expand the window and display the transactions in bar graph form. This graph displays the current number of bytes (or packet) for all terminals and for the specified time period (60 seconds or 60 minutes).



TX Bytes 4.084
TX Packets 257
RX Bytes 5.216
RX Packets 256

<Local>
099 / IP: 192.42.172.100
Started: 25/03/1999 18:14
18:15:14

Figure 3-6. Online Graph Display

Each bar segment corresponds to 1 second or 1 minute depending on the selected view. Right-click over the graph or choose *On Line* from the *Options* menu to select the graph parameters.

– decrease (-) the vertical scale unit.

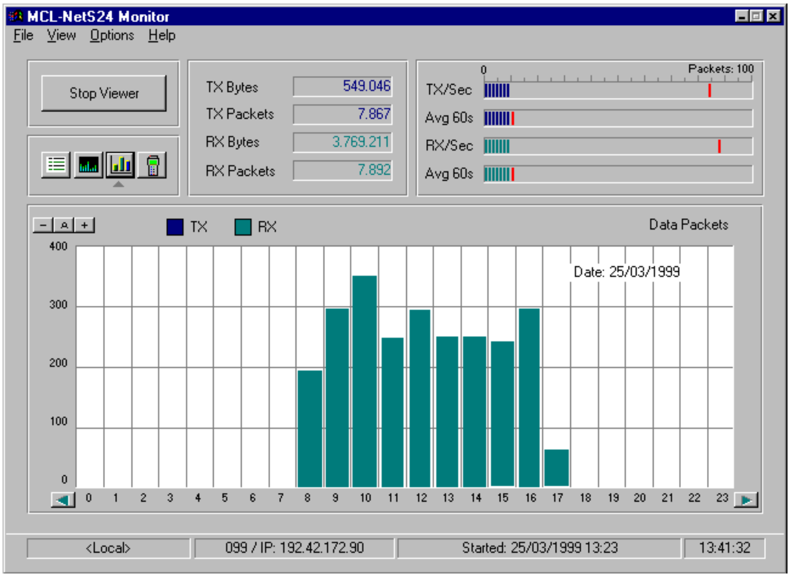
+ increase (+) the vertical scale unit.

A automatic adjustment of the vertical scale unit.

◀ or ▶ move to the next or previous time period.

Statistics Graph Display

Click the *Statistics Graph Display* button to expand the window and display the statistics in bar graph form. This graph displays the specified statistics for terminal(s) for the specified time period. The graph can be viewed in either 24 hour view or 31 days view (Refer to *Statistics Setup* on page 3-9 for more information).



TX Bytes 549.046
TX Packets 7.867
RX Bytes 3.769.211
RX Packets 7.892

Date: 25/03/1999

<Local>
099 / IP: 192.42.172.90
Started: 25/03/1999 13:23
13:41:32

Figure 3-7. Statistics Graph Display

– decrease (-) the vertical scale unit.

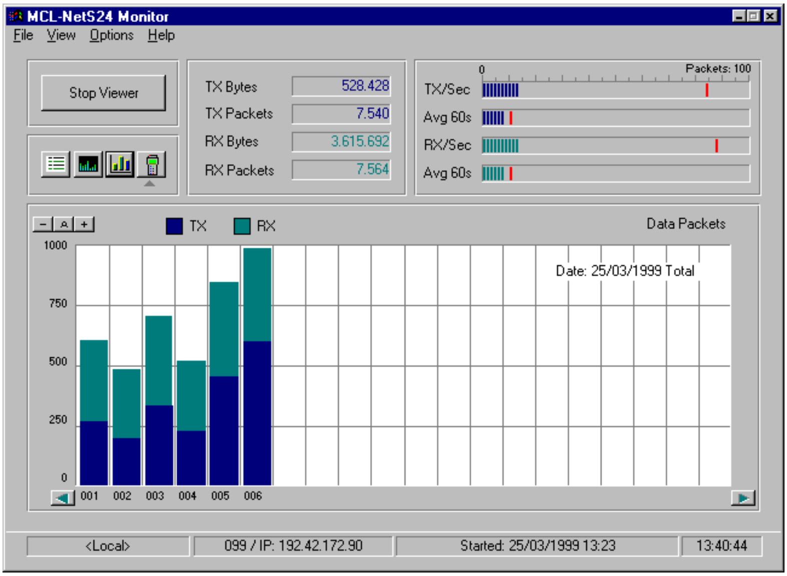
+ increase (+) the vertical scale unit.

A automatic adjustment of the vertical scale unit.

◀ or ▶ move to the next or previous time period.

Terminals Statistics Graph Display

Click the *Terminals Statistics Graph Display* button to expand the window and display the statistics in bar graph form. This graph displays the specified statistics for terminal(s) for the specified time period. Each bar represents a terminal. (Refer to *Statistics Setup* on page 3-9 for more information.)



TX Bytes 528.428
TX Packets 7.540
RX Bytes 3.615.692
RX Packets 7.564

<Local>
099 / IP: 192.42.172.90
Started: 25/03/1999 13:23
13:40:44

Figure 3-8. Terminal Statistics Graph Display

− decrease (-) the vertical scale unit.

+ increase (+) the vertical scale unit.

A automatic adjustment of the vertical scale unit.

◀ or ▶ move to the next or previous group of terminals.

Statistics Setup

The *Statistics Setup* dialog box can be accessed from the *Options* menu and choosing *Statistics* or by right clicking on the Statistic graph.

The Statistics Setup dialog box is different depending on the selection of the *Period Statistics* or the *Terminal Statistics*.

Period Statistic

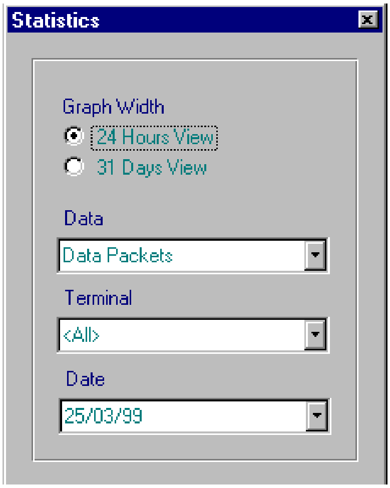


Figure 3-9. Period Statistics Dialog Box

Graph Width Determines the scope of the graph. Shows activity in 24 hour increments or in 31 day increments.

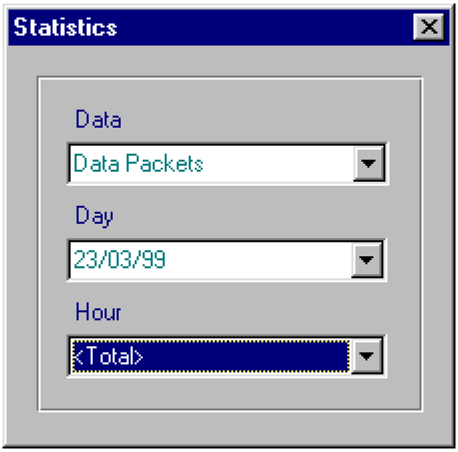
Data Determines the type of statistics to display. Available options are:

- Connect/Disconnect
- Data Packets
- File Packets
- ACK Packets
- NACK Packets
- Busy Packets
- Reject Packets
- Abort Packets
- Retry Packets.

Terminal Determines which terminals to display.

Date Determines which date to display.

Terminal Statistics



Statistics

Data

Data Packets

Day

23/03/99

Hour

<Total>

Figure 3-10. Terminal Statistics Dialog Box

Data Determines the type of statistics to display. Available options are:

- ♦ Connect/Disconnect
- ♦ Data Packets
- ♦ File Packets
- ♦ ACK Packets
- ♦ NACK Packets
- ♦ Busy Packets
- ♦ Reject Packets
- ♦ Abort Packets
- ♦ Retry Packets.

Day Determines which date to display.

Hour Determines which hour to display.

Index

Tell Us What You Think...

We’d like to know what you think about this Manual. Please take a moment to fill out this questionaire and fax this form to: (516) 738-3318, or mail to:

Symbol Technologies, Inc.
One Symbol Plaza   M/S B-4
Holtsville, NY 11742-1300
Attn: Technical Publications Manager

IMPORTANT: If you need product support, please call the appropriate customer support number provided. Unfortunately, we cannot provide customer support at the fax number above.

User’s Manual Title: ____
(please include revision level)

How familiar were you with this product before using this manual?

□ Very familiar   □ Slightly familiar   □ Not at all familiar

Did this manual meet your needs? If not, please explain. ______________________
____

What topics need to be added to the index, if applicable? ______________________
____

What topics do you feel need to be better discussed? Please be specific. ____________
____

What can we do to further improve our manuals? ______________________________
____
____

Thank you for your input—We value your comments.

MCL-NetS24 Version 2 User's Guide

70-33348-02
Revision A — June 1999

Symbol Technologies, Inc. One Symbol Plaza, Holtsville N.Y. 11742-1300